

**PROSES PENENTUAN JENDELA FFT (*FAST FOURIER TRANSFORM*)  
UNTUK 64 TITIK DAN *CYCLIC PREFIX 1/4* PADA SISTEM  
OFDM (*ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING*)  
16-QAM (*QUADRATURE AMPLITUDO MODULATION*)**

**SKRIPSI  
KONSENTRASI TEKNIK TELEKOMUNIKASI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh :

**MUHAMMAD RIAD**

**NIM. 0710632010**

**KEMENTERIAN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG**

**2012**

**PROSES PENENTUAN JENDELA FFT (*FAST FOURIER TRANSFORM*)  
UNTUK 64 TITIK DAN *CYCLIC PREFIX 1/4* PADA SISTEM  
OFDM (*ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING*)  
16-QAM (*QUADRATURE AMPLITUDO MODULATION*)**

**SKRIPSI  
KONSENTRASI TEKNIK TELEKOMUNIKASI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun oleh :  
**MUHAMMAD RIAD**  
**NIM. 0710632010**

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

**Ir. Erfan Achmad Dahlan, MT.**  
**NIP. 19530714 198203 1 003**

**Dwi Fadila Kurniawan, ST., MT.**  
**NIP. 19720630 200003 1 002**

**LEMBAR PENGESAHAN**

**PROSES PENENTUAN JENDELA FFT (FAST FOURIER TRANSFORM )  
UNTUK 64 TITIK DAN CYCLIC PREFIX 1/4 PADA SISTEM  
OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING)  
16-QAM (QUADRATURE AMPLITUDO MODULATION )**

**SKRIPSI  
KONSENTRASI TEKNIK TELEKOMUNIKASI**

Disusun Oleh:

**MUHAMMAD RIAD.**

**NIM. 0710632010**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 03 Januari 2012

**Majelis Penguji :**

**Ir. Wahyu Adi Priyono, MSc.**  
**NIP. 19600518 198802 1 001**

**Ir. Sigit Kusmaryanto, MT.**  
**NIP. 19700310 199412 1 001**

**Ir. Endah Budi P., MT.**  
**NIP. 19621116 198903 2 002**

Mengetahui,  
Ketua Jurusan Teknik Elektro

**Dr. Ir. Sholeh Hadi Pramono, MS**  
**NIP. 19580728 198701 1 001**

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT karena atas berkat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Skripsi ini dengan judul **“Proses Penentuan Jendela FFT (*Fast Fourier Transform*) Untuk 64 Titik dan Cyclic Prefix 1/4 Pada Sistem OFDM (*Orthogonal Frequency Division Multiplexing*) 16-QAM (*Quadrature Amplitudo Modulation*)”**. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Ir. Erfan Achmad Dahlan, MT., dan Bapak Dwi Fadila Kurniawan, ST., MT., sebagai dosen pembimbing atas saran, konsultasi, kesabaran dan waktu dalam pengerjaan hingga penyelesaian tugas akhir ini. Tidak terkecuali, ucapan terima kasih penulis sampaikan kepada :

1. Ayahanda yang selalu menjadi inspirasi.
2. Ibunda tercinta yang selalu mendukung serta kasih sayang yang telah diberikan.
3. Istriku tersayang serta kakak dan adikku yang selalu mendo'akan dan memberi semangat demi kesuksesanku.
4. Bapak Dr. Ir. Sholeh Hadi Pramono, MS. selaku Ketua Jurusan Teknik Elektro.
5. Bapak M. Azis Muslim, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro.
6. Bapak Ali Mustofa, ST., MT, selaku Ketua Kelompok Dosen dan Keahlian Teknik Telekomunikasi.
7. Bapak Mohammad Rif'an, ST., MT., selaku Ketua Program Studi
8. Seluruh Dosen dan Karyawan Jurusan Teknik Elektro.
9. Sahabat-sahabatku, Fakhriy Hario, Rangga Dwi S, Guruh WS, Nurina Evawani, Andri Fahlevi, tanpa kalian tidak akan ada skripsi ini.
10. Seluruh rekan-rekan mahasiswa Jurusan Teknik Elektro atas kebersamaan dan pengetahuannya. Dan semua pihak yang tidak dapat penulis sebutkan satu-persatu.

Dalam penyusunan skripsi ini, penyusun menyadari akan adanya kekurangan. Oleh karena itu, penyusun sangat mengharapkan kritik dan saran yang membangun untuk kelengkapan dan kesempurnaan skripsi ini. Penyusun berharap semoga skripsi ini dapat bermanfaat khususnya bagi rekan-rekan mahasiswa.

Malang, Desember 2011

Penulis

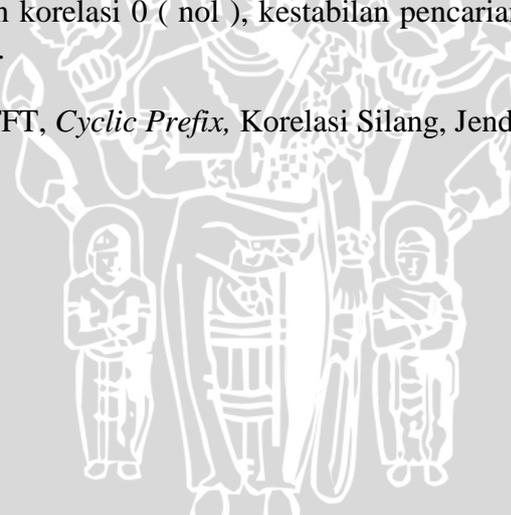


## ABSTRAK

Riad, Muhammad. 2011. Proses Penentuan Jendela FFT (*Fast Fourier Transform*) Untuk 64 Titik dan *Cyclic Prefix* 1/4 Pada Sistem OFDM (*Orthogonal Frequency Division Multiplexing*) 16-QAM (*Quadrature Amplitudo Modulation*). Jurusan Teknik Elektro. Fakultas Teknik. Universitas Brawijaya Malang. Pembimbing (1). Ir. Erfan Achmad Dahlan, MT., (2). Dwi Fadila Kurniawan, ST., MT.

OFDM (*Orthogonal Frequency Division Multiplexing*) adalah teknik modulasi menggunakan *subcarrier orthogonal* antar sub-kanal. Jendela FFT berisi informasi yang ditransmisikan dari sisi pemancar ke sisi penerima. Tujuan dari penelitian ini adalah menganalisis sebagian tahapan yang terjadi pada sistem OFDM yaitu mengenai keberhasilan pencarian jendela FFT (*Fast Fourier Transform*), periode waktu pencarian jendela FFT, kemungkinan kegagalan yang terjadi dalam penentuan jendela FFT dan kemungkinan adanya kesalahan (*error*) dalam penentuan jendela FFT. *Software* simulasi khusus yang telah dibuat menggunakan Delphi 7.0 untuk melakukan penelitian ini. Hasil dari penelitian ini memperlihatkan bahwa jika dengan menggunakan metode korelasi, terdapat pengaruh derau dari kanal AWGN terhadap waktu pencarian jendela FFT dan tingkat keberhasilan pencarian jendela FFT. Dibutuhkan daya 0.005 Watt untuk dapat stabil dalam keberhasilan pencarian jendela FFT, batas Kuantitatif SNR(dB) = 19.5 dB menghasilkan nilai koefisien korelasi 0 ( nol ), kestabilan pencarian jendela FFT dengan batas minimal 29,5 dB SNR.

Kata kunci: OFDM, IFFT, FFT, *Cyclic Prefix*, Korelasi Silang, Jendela FFT.



## DAFTAR ISI

Kata Pengantar.....	i
Abstrak .....	ii
Daftar Isi .....	iii
Daftar Gambar .....	v
Daftar Tabel .....	vi
Daftar Singkatan .....	vii

### BAB I PENDAHULUAN

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan penelitian .....	3
1.5 Sistematika Penulisan .....	4

### BAB II TINJAUAN PUSTAKA

2.1 Konsep Dasar OFDM .....	5
2.2 Sistem Transceiver OFDM .....	7
2.3 IFFT dan FFT .....	8
2.4 FFT DIF <i>Radix-2</i> .....	11
2.5 Diagram Kupu-kupu <i>Gentlemen-Sande</i> .....	13
2.6 IFFT DIF <i>Radix-2</i> .....	15
2.7 <i>Delay Spread</i> .....	16
2.8 Interval Pengaman dan <i>Cyclic Prefix</i> .....	17
2.9 16 QAM .....	18
2.10 AWGN .....	19
2.11 Fungsi Korelasi Silang ( <i>Cross Correlation Function, CCF</i> ) .....	22

### BAB III METODOLOGI PENELITIAN

3.1 Studi Literatur .....	24
---------------------------	----

3.2 Pengumpulan Data .....	24
3.2.1 Data Sekunder .....	24
3.2.2 Simulasi Sistem .....	25
3.3 Analisis Data.....	26
3.4 Pengambilan Kesimpulan .....	26

#### BAB IV HASIL SIMULASI DAN ANALISIS

4.1 Perancangan Simulasi Sistem .....	27
4.2 Pengujian <i>Software</i> Simulasi .....	28
4.2.1 Tampilan <i>Software</i> Simulasi.....	29
4.2.2 Pembangkit Masukan IFFT.....	30
4.2.3 IFFT dan FFT.....	33
4.2.4 Penambahan <i>Cyclic Prefix</i> .....	35
4.2.5 Paralel ke Serial .....	37
4.2.6 Kanal AWGN.....	38
4.2.7 Jendela FFT.....	39
4.3 Analisis Hasil Percobaan .....	44
4.3.1 Pengaruh Daya Derau Terhadap Waktu Pencarian Jendela FFT....	45
4.3.2 Pengaruh Daya Derau Terhadap Nilai Koefisien Korelasi .....	47
4.3.1 Pengaruh Daya Derau Terhadap Keberhasilan Pencarian Jendela..	50
4.3.4 Pengaruh Daya Derau Terhadap Kecermatan Pencarian Jendela ...	51
4.3.5 Pengaruh Batas Koefisien Korelasi Terhadap Waktu Pencarian Jendela FFT .....	53

#### BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan .....	56
5.2 Saran .....	56

DAFTAR PUSTAKA .....	57
----------------------	----

LAMPIRAN .....	58
----------------	----



## DAFTAR GAMBAR

Gambar 2.1 Perbandingan penggunaan <i>bandwidth</i> pada FDM dan OFDM .....	5
Gambar 2.2 Diagram <i>Transceiver</i> OFDM .....	6
Gambar 2.3 Model konseptual pembangkitan sinyal OFDM empat sub-kanal dan spektrumnya.....	8
Gambar 2.4 Diagram blok FFT .....	10
Gambar 2.5 Diagram kupu-kupu <i>Gentlemen-Sande</i> .....	13
Gambar 2.6 Diagram alir FFT DIF <i>Radix-2</i> 8 Titik .....	14
Gambar 2.7 Diagram blok IFFT .....	15
Gambar 2.8 <i>Delay Spread</i> pada kanal-jamak .....	16
Gambar 2.9 Penggunaan <i>Cyclic Prefix</i> .....	17
Gambar 2.10 Konstelasi sinyal 16-QAM.....	18
Gambar 2.11 Pemodelan AWGN pada kanal transmisi.....	20
Gambar 2.12 Fungsi Korelasi .....	22
Gambar 3.1 Diagram alir Pencarian Jendela FFT .....	24
Gambar 4.1 Diagram blok sistem OFDM pada <i>software</i> simulasi .....	26
Gambar 4.2 Tampilan <i>software</i> simulasi .....	28
Gambar 4.3 Data masukan IFFT yang dibangkitkan .....	29
Gambar 4.4 Tampilan data IFFT .....	32
Gambar 4.5 Tampilan data penambahan <i>Cyclic Prefix</i> .....	34
Gambar 4.6 Tampilan data serial pada P/S di bagian pengirim .....	36
Gambar 4.7 Tampilan data informasi setelah terpengaruh kanal AWGN .....	37
Gambar 4.8 Tampilan sinyal pada kanal AWGN .....	38
Gambar 4.9 Tampilan cuplikan awal .....	39
Gambar 4.10 Tampilan pencarian jendela FFT .....	39
Gambar 4.11 Tampilan penghitung proses pencarian jendela FFT .....	40
Gambar 4.12 Tampilan hasil pencarian jendela FFT.....	43
Gambar 4.13 Pengaruh daya derau terhadap waktu pencarian jendela FFT.....	45
Gambar 4.14 Pengaruh daya derau terhadap nilai koefisien korelasi .....	47
Gambar 4.15 Pengaruh daya derau terhadap hasil koefisien korelasi .....	50
Gambar 4.16 Ketepatan nilai koefisien korelasi .....	52
Gambar 4.17 Pengaruh batas koefisien korelasi terhadap waktu pencarian jendela FFT .....	54

**DAFTAR TABEL**

Tabel 2.1	Indeks Masukan dan Keluaran FFT DIF <i>Radix-2</i> 8 Titik .....	14
Tabel 4.1	Hasil Pembangkitan data masukan IFFT secara acak .....	30
Tabel 4.2	Pemetaan 16-QAM .....	32
Tabel 4.3	Keluaran IFFT pada <i>software</i> simulasi .....	33
Tabel 4.4	Data hasil penambahan <i>Cyclic Prefix</i> .....	35
Tabel 4.5	Data yang akan dikorelasikan .....	40
Tabel 4.6	Hasil percobaan pengaruh daya derau yang bervariasi terhadap nilai koefisien korelasi yang dihasilkan .....	44
Tabel 4.7	Hasil percobaan pengaruh daya derau terhadap nilai koefisien korelasi .....	46
Tabel 4.8	Hasil percobaan tingkat keberhasilan pencarian jendela FFT .....	50
Tabel 4.9	Hasil percobaan tingkat kecermatan pencarian jendela FFT .....	51
Tabel 4.10	Hasil percobaan pengaruh batas koefisien korelasi terhadap waktu pencarian jendela FFT .....	53



## DAFTAR SINGKATAN

ACF	Auto Correlation Function
AM	Amplitudo Modulation
ASK	<i>Amplitude Shift Keying</i>
AWGN	Adaptive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CCF	Cross Correlation Function
DAC	Digital To Analogue Converter
DIF	Decimation In Frequency
DIT	Decimation In Time
GI	Guard Interval
IFFT	Inverse Fast Fourier Transform
ISI	Inter Symbol Interference
FM	Frequency Modulation
FDM	Frequency Division Multiplexing
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
OFDM	Orthogonal Frequency Division Multiplexing
PS	Parallel to Serial
PSK	Phase Shift Keying
SNR	Signal Noise Ratio
QAM	Quadrature Amplitudo Modulation
QPSK	Quadrature Phase Shift Keying

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi di era globalisasi seperti sekarang ini, menyebabkan perkembangan yang pesat pula di bidang telekomunikasi. Perkembangan teknologi telekomunikasi yang semakin canggih akan semakin memudahkan manusia untuk dapat saling bertukar informasi dengan cepat dan tepat sesuai yang diharapkan. Sistem komunikasi nirkabel merupakan sistem komunikasi yang berkembang sangat pesat dewasa ini, yang memungkinkan manusia dapat berkomunikasi dimanapun dan kapanpun.

Kebutuhan akan komunikasi kini tidak terbatas hanya pada komunikasi suara saja namun juga pada komunikasi data, aplikasi multimedia, dan lain-lain, sehingga diperlukan sistem telekomunikasi yang dapat mengirimkan informasi dengan *bit rate* yang sangat tinggi dan BER (*Bit Error Rate*) yang sangat rendah dengan menggunakan lebar pita (*bandwidth*) secara efisien. *Bit rate* yang sangat tinggi dan BER yang sangat rendah diperlukan agar berbagai jenis informasi baik suara, data dan video dapat sampai ke sisi penerima dengan baik, sedangkan penggunaan *bandwidth* seminimal mungkin diperuntukan karena *bandwidth* merupakan sumber daya yang terbatas dan mahal sehingga harus digunakan secara efisien. Proses tersebut bukan hal yang mudah mengingat semakin tinggi *bit rate* mengakibatkan semakin besar *bandwidth* yang dibutuhkan, sehingga dibutuhkan suatu evolusi teknologi yang dapat menyediakan layanan dengan *bit rate* tinggi dan efisiensi pemakaian *bandwidth*, salah satu teknologi tersebut adalah teknologi OFDM (*Orthogonal Frequency Division Multiplexing*).

Sistem OFDM merupakan salah satu solusi sistem komunikasi nirkabel yang mempunyai beberapa kelebihan, diantaranya yaitu mampu menyediakan layanan dengan *bit rate* yang tinggi namun tetap efisien dalam pemakaian *bandwidth* dengan BER yang rendah. OFDM merupakan teknik transmisi dengan sinyal *carrier* yang saling *orthogonal*. Pada OFDM digunakan beberapa frekuensi *carrier* yang saling *orthogonal* sehingga memungkinkan rentang frekuensi antar *carrier* dapat saling *overlap* tanpa menimbulkan interferensi satu sama lain. OFDM menggunakan

beberapa *carrier* yang saling *orthogonal* sehingga efisien dalam pemanfaatan *bandwidth*.

*Bit rate* dan efisiensi dalam pemakaian *bandwidth* pada sistem OFDM dapat dilakukan dengan cara membagi *bandwidth* yang lebar menjadi beberapa *bandwidth* yang lebih sempit. Sistem OFDM telah digunakan sebagai standar operasi di Eropa dan Australia yaitu pada proyek televisi digital yaitu DAB (*Digital Audio Broadcasting*), DVB (*Digital Video Broadcatsing*), WiFi (*Wireless Fidelity*), MC-CDMA (*Multicarrier Code Division Multiple Access*), dan diterapkan juga pada WiMAX (*Worldwide Interoperability for Microwave Access*).

Skripsi ini akan mengkaji sebagian tahapan yang terjadi pada sistem OFDM yaitu mengenai pencarian jendela FFT (*Fast Fourier Transform*) pada penerima sistem OFDM. Data yang diperoleh dari parameter tersebut dapat digunakan untuk menganalisis tentang pengaruh penambahan interval pengaman berupa *Cyclic Prefix* dan pengaruh besarnya derau dari kanal AWGN (*Adaptive White Gaussian Noise*) terhadap proses pencarian jendela FFT pada sisi penerima OFDM yang menyangkut beberapa hal yaitu diantaranya mengenai keberhasilan penentuan jendela FFT, periode waktu pencarian jendela FFT, kemungkinan kegagalan yang terjadi dalam pencarian jendela FFT dan kemungkinan adanya kesalahan (*error*) dalam penentuan jendela FFT.

## 1.2 Rumusan Masalah

Berdasarkan penjelasan yang terdapat pada latar belakang, maka perumusan masalah skripsi ini dapat diuraikan sebagai berikut:

1. Bagaimana proses penentuan jendela FFT pada sisi penerima OFDM dengan penambahan *Cyclic Prefix*.
2. Bagaimana pengaruh adanya derau dari kanal AWGN terhadap keberhasilan penentuan jendela FFT dan periode waktu.
3. Bagaimana kemungkinan *error* yang terjadi dalam penentuan jendela FFT.

### 1.3 Batasan Masalah

Berdasarkan perumusan masalah di atas, maka akan dibuat batasan-batasan masalah yang akan menspesifikasikan hal-hal yang akan dibahas. Batasan-batasan masalah ini dimaksudkan agar skripsi yang dibuat dapat fokus pada masalah yang diteliti.

Batasan-batasan masalah tersebut adalah sebagai berikut :

1. Data masukan merupakan data kompleks masukan IFFT (*Inverse Fast Fourier Transform*) yang dibangkitkan secara acak.
2. Modul IFFT/FFT dibatasi untuk *Radix-2* yang disimulasikan untuk jumlah titik 64 titik.
3. Terdiri dari 64 sub-kanal.
4. Pemetaan yang digunakan adalah 16-QAM.
5. Simulasi OFDM menggunakan interval pengaman *Cyclic Prefix*  $\frac{1}{4}$ .
6. Kanal transmisi hanya dipengaruhi oleh AWGN.
7. Daya derau yang digunakan adalah 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt.
8. Batasan koefisien korelasi yang digunakan adalah 0.8, 0.9, 1.
9. Simulasi sistem OFDM menggunakan konfigurasi yang sederhana yang bekerja pada bidang dasar dari beberapa bagian saja di sisi pengirim dan penerima sistem OFDM.
10. Simulasi sistem OFDM dibuat menggunakan *software* Delphi 7.0.

### 1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah untuk menganalisis proses penentuan jendela FFT untuk 64 titik dan *cyclic prefix*  $\frac{1}{4}$  pada sistem OFDM (*Orthogonal Frequency Division Multiplexing*) 16-QAM.

## 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut :

### **BAB I   Pendahuluan**

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan.

### **BAB II   Dasar Teori**

Bab ini membahas tentang konsep dasar OFDM, konfigurasi sederhana OFDM, IFFT/FFT pada sistem OFDM, *Cyclic Prefix* dan teori yang mendukung dalam penelitian.

### **BAB III  Metodologi**

Bab ini menjelaskan tentang metode yang digunakan dalam penulisan yang meliputi studi pustaka, metode pengambilan data, metode perhitungan dan analisa data, dan metode pengambilan kesimpulan.

### **BAB IV  Perancangan Simulasi dan Analisa Data Hasil Simulasi**

Bab ini menjelaskan tentang gambaran umum simulasi, perancangan simulasi beberapa bagian dari sistem OFDM dan diagram alir simulasi serta tahapan-tahapan yang terjadi pada simulasi beberapa bagian dari sistem OFDM dan analisis mengenai proses pencarian jendela FFT pada sisi penerima dengan metode korelasi yang terjadi pada sistem OFDM berdasarkan data hasil percobaan.

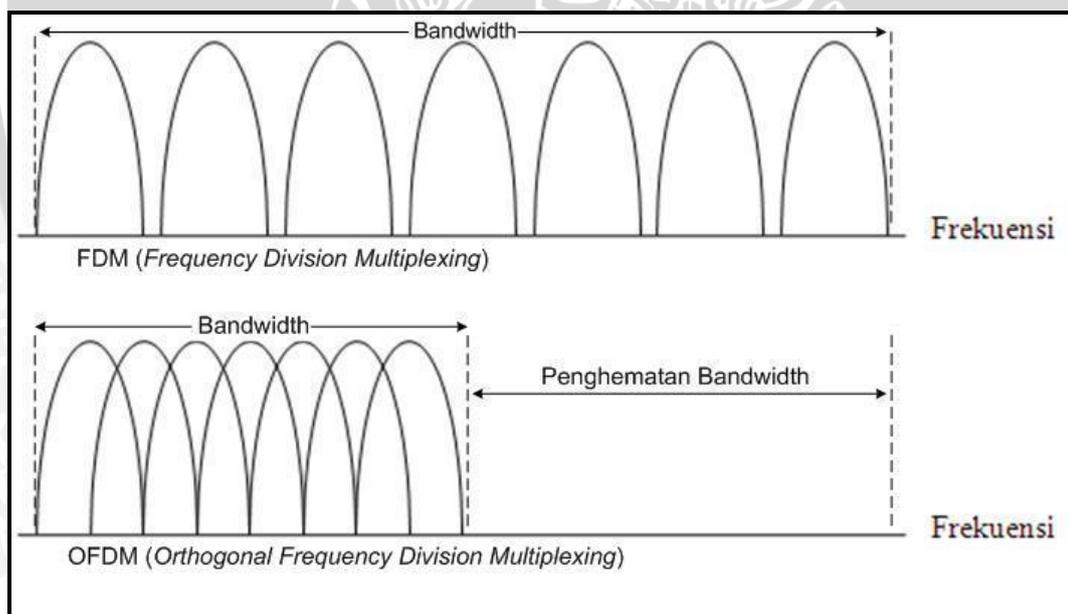
### **BAB V   Kesimpulan dan Saran**

Bab ini berisi kesimpulan yang didapatkan dan saran yang dapat digunakan untuk membuat penelitian yang lebih baik.

## BAB II DASAR TEORI

### 2.1 Konsep Dasar OFDM (*Orthogonal Frequency Division Multiplexing*)

OFDM merupakan bentuk pengembangan dari penjamakan frekuensi FDM (*Frequency Division Multiplexing*) dengan teknik modulasi *multicarrier* yaitu teknik transmisi yang menggunakan beberapa frekuensi *subcarrier* yang dipisahkan oleh *bandwidth* tertentu supaya tidak terjadi interferensi sehingga memerlukan *bandwidth* yang cukup besar. Sistem OFDM pada dasarnya membagi aliran *bit* dengan *bit rate* tinggi menjadi beberapa aliran *bit* dengan *bit rate* yang lebih rendah. Penggunaan *bit rate* yang lebih rendah menyebabkan periode simbol akan bertambah sehingga *delay* dapat ditekan pada penerima. *Bit-bit* tersebut kemudian dikirimkan secara bersamaan dengan beberapa sub-kanal *carrier* yang saling *orthogonal*. Hal ini menunjukkan bahwa OFDM lebih efisien dalam penggunaan *bandwidth* dibandingkan dengan FDM seperti pada Gambar 2.1



**Gambar 2.1** Perbandingan penggunaan lebar pita (*bandwidth*) pada FDM dan OFDM  
(Sumber: Faisal Muhammad, 2009:12)

Gambar 2.1 mengilustrasikan perbedaan antara teknik *multicarrier non overlap* konvensional dan teknik modulasi *multicarrier orthogonal*, teknik ini dapat menghemat hampir 50% *bandwidth*. Ortogonalitas didapat dengan mengatur jarak antar *carrier*

sebesar  $1/T$ , dimana  $T$  adalah periode simbol. Penggunaan frekuensi *subcarrier* yang saling tegak lurus (*orthogonal*) merupakan salah satu konsep dasar OFDM. Istilah *orthogonal* mengisyaratkan hubungan matematis, dua buah vektor dikatakan saling tegak lurus jika perkalian titik (*dot product*) kedua vektor tersebut adalah nol atau besarnya sudut yang dibentuk kedua vektor tersebut adalah  $90^0$ . Dua buah sinyal dikatakan saling *orthogonal* jika kedua sinyal tersebut tidak saling mengganggu atau tidak menimbulkan interferensi. Ortogonalitas pada OFDM mengandung makna hubungan matematis antara frekuensi-frekuensi yang digunakan.

Secara matematis, untuk membuat setiap sinyal *orthogonal* adalah dengan membuat luas area positif sama dengan luas area negatif atau hasil integral dari sinyal tersebut adalah nol.

$$f(t) = \sin(n\omega t) \sin(m\omega t) \quad (2.1)$$

Persamaan 2.1 dapat diturunkan menjadi:

$$f(t) = \frac{1}{2} \{ \cos[(m-n)\omega t] - \frac{1}{2} \{ \cos[(m+n)\omega t] \} \quad (2.2)$$

Dimana:

$n$  dan  $m$  = konstanta sinyal yang saling tegak lurus  
 $\omega = 2\pi f$  ;  $f$  = frekuensi sinyal *carrier* (Hertz)

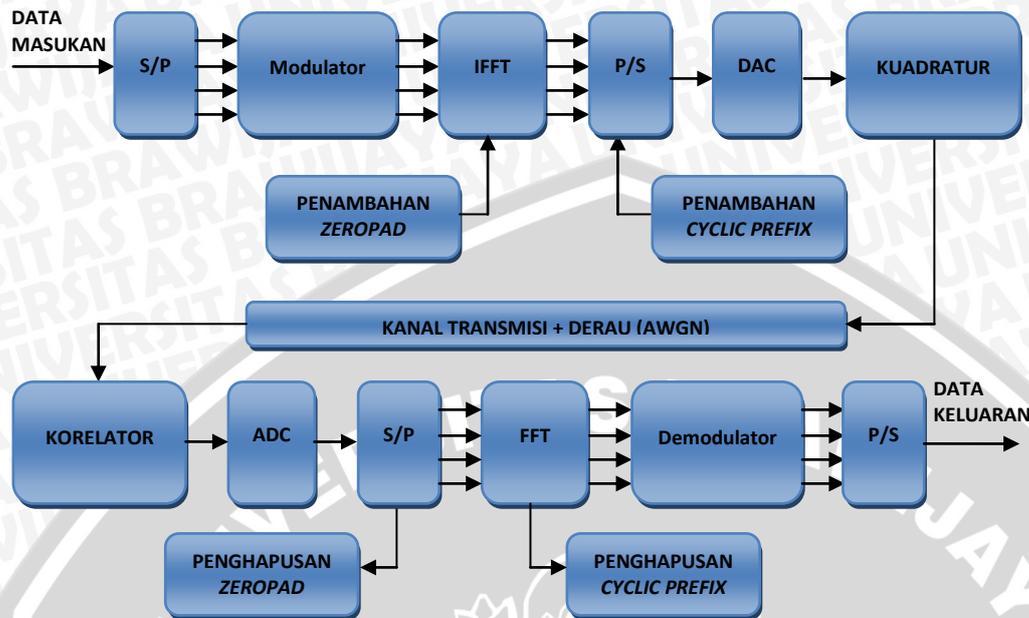
Luas daerah (LA) dinyatakan dengan persamaan 2.3:

$$LA = \int_0^{2\pi} \frac{1}{2} \{ \cos[(m-n)\omega t] - \int_0^{2\pi} \frac{1}{2} \{ \cos[(m+n)\omega t] \} \quad (2.3)$$

Jika Luas daerah adalah 0, maka  $\sin(n\omega t)$  dan  $\sin(m\omega t)$  saling *orthogonal* untuk nilai  $m$  dan  $n$  adalah *integer*.

## 2.2 Sistem Transceiver OFDM

Gambar 2.2 menunjukkan blok diagram *transceiver* dari sistem OFDM.



**Gambar 2.2 Blok Diagram *Transceiver* OFDM**

(Sumber: Mustafa Ergen, 2009)

Pada sisi pengirim *bit-bit* informasi dimodulasi dengan salah satu teknik modulasi digital (16-QAM), setelah dimodulasi dengan setiap *subcarrier orthogonal*. *Bit-bit* informasi tersebut menjadi sinyal serial yang termodulasi. Kemudian sinyal serial tersebut diubah ke bentuk sinyal paralel, masing-masing sinyal paralel masuk ke blok IFFT (*Inverse Fast Fourier Transform*). Sinyal yang keluar dari blok IFFT merupakan sinyal *multicarrier* berupa simbol OFDM yang merupakan bilangan kompleks dikonversi menjadi bentuk serial dengan cara memisahkan antara bagian *real* dan *imajiner*. Sebelum dikonversi ke bentuk analog, sinyal tersebut ditambahkan *guard interval (cyclic prefix)*. Penambahan *guard interval* berguna untuk menjaga sinyal agar dapat meminimalisasi ISI (*Inter Symbol Interference*). Kemudian bagian *real* dan *imajiner* tersebut dikonversi menjadi bentuk analog dengan menggunakan pengubah digital ke analog (*Digital To Analogue Converter, DAC*). Setelah itu, pada kuadratur simbol OFDM tersebut dimodulasi dan kemudian dijumlahkan. Keluaran dari kuadratur berupa sinyal informasi dikirimkan melalui kanal transmisi (AWGN) sehingga dapat diterima oleh bagian penerima. Pada sisi penerima dilakukan proses yang berkebalikan dengan sisi pengirim sehingga pada akhirnya didapatkan informasi berupa *bit* terima.

### 2.3 IFFT dan FFT

Teknik transmisi OFDM menggunakan beberapa *subcarrier* yang saling *orthogonal* yang berasal dari pembangkitan beberapa osilator frekuensi. Pembangkitan osilator frekuensi akan banyak menimbulkan kompleksitas dan kerumitan dalam desain implementasi perangkat keras. Solusi untuk mengatasi kerumitan yang diakibatkan oleh banyaknya *modulator* dan *demodulator analog* yang harus digunakan pada sistem OFDM, maka pada tahun 1971 *Weinstein* dan *Ebert* memperkenalkan konsep baru untuk menggantikan *modulator* dan *demodulator analog* yang sangat banyak dan rumit pada OFDM dengan menggunakan IDFT dan DFT (Hanzo dan Keller, 2006).

Masukan IFFT/FFT pada sistem OFDM ini merupakan bilangan kompleks. Simbol QAM yang merupakan masukan IFFT di bagian pengirim dan keluaran FFT di bagian penerima merupakan bilangan kompleks, begitu juga simbol OFDM yang merupakan keluaran IFFT dan masukan FFT merupakan bilangan kompleks. Hal ini membuat komputasi bilangan kompleks merupakan dasar dari proses IFFT/FFT.

Pada IFFT/FFT dilakukan proses komputasi bilangan kompleks baik penjumlahan, pengurangan, perkalian dan pembagian. Aturan-aturan proses komputasi tersebut adalah sebagai berikut :

$$(a + j b) + (c + j d) = (a + c) + j (b + d) \quad (2.4)$$

$$(a + j b) - (c + j d) = (a - c) + j (b - d) \quad (2.5)$$

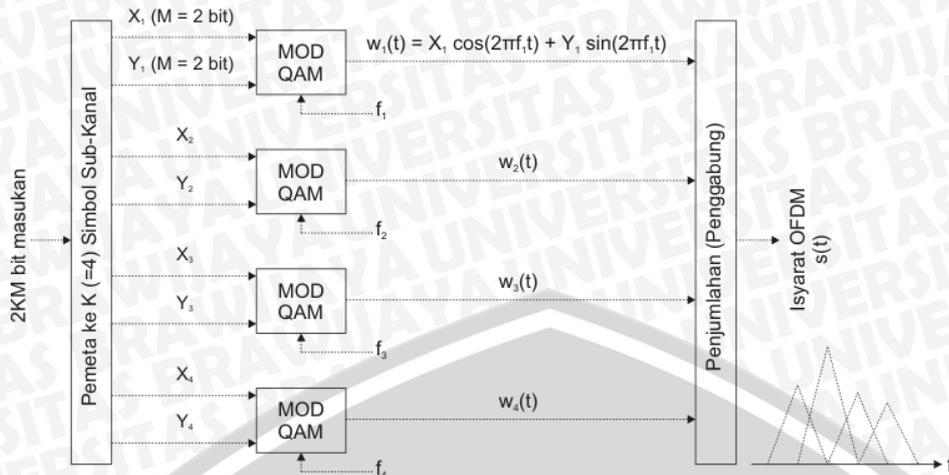
$$(a + j b) * (c + j d) = (a*c - b*d) + j (a*d + b*c) \quad (2.6)$$

$$(a + j b) / e = (a / e) + j (b / e) \quad (2.7)$$

Berikut adalah penjelasan mengenai penggantian modulator/demodulator analog dengan IDFT/DFT pada sistem OFDM (Setiyanto dan Mulyana, 2008). Andaikan *bit-bit* informasi dipilah-pilah menjadi simbol-simbol  $m_1(t), m_2(t), m_3(t), \dots, m_k(t)$  yang berturut-turut dibawa oleh frekuensi  $f_1, f_2, f_3, \dots, f_k$  maka sinyal QAM-FDM adalah

$$s_{FDM}(t) = \sum_{k=1}^K \text{Re}[ m_k(t) e^{j2\pi f_k t} ] \quad (2.8)$$

Jika setiap simbol memiliki durasi  $T$  dan dibuat  $f_k = k/T$ , maka masing-masing suku pada persamaan (2.8), atau masing-masing sub-kanal menjadi saling *orthogonal*. Model konseptual pembangkitan OFDM empat sub-kanal dengan modulasi 16-QAM diperlihatkan pada Gambar 2.3. (Setiyanto dan Mulyana, 2008).



Gambar 2.3 Model konseptual pembangkitan sinyal OFDM empat sub-kanal dan spektrumnya (Sumber: Setiyanto dan Mulyana, 2008)

Substitusi  $f_k = k/T$  dan menyertakan bagian *imajiner* dalam penjumlahan pada persamaan (2.8) menghasilkan

$$s(t) = \sum_{k=1}^K m_k(t) e^{j2\pi kt/T} \tag{2.9}$$

Andaikan dalam durasi  $T$ ,  $m_k(t)$  maupun  $s(t)$  dicuplik-cuplik dengan interval  $T_s = T/K$ , maka persamaan (2.6) dapat ditulis menjadi

$$s(nT_s) = \sum_{k=1}^K m_k(nT_s) e^{j2\pi knT_s/(KT_s)} = \sum_{k=1}^K m_k(nT_s) e^{j2\pi kn/K} \tag{2.10}$$

Namun, untuk setiap  $k$ , dalam durasi  $T = KT_s$ , nilai  $m_k(nT_s)$  adalah konstan sehingga perbedaan nilai  $m_k(nT_s)$  adalah berdasar  $k$  bukan  $n$ . Oleh karena itu persamaan (2.10) tidak lain adalah bentuk IDFT. (Setiyanto dan Mulyana, 2008). DFT (*Discrete Fourier Transform*) merupakan metode transformasi yang mengubah sinyal diskrit dari kawasan waktu menjadi kawasan frekuensi sedangkan IDFT (*Inverse Discrete Fourier Transform*) adalah proses kebalikan dari DFT.. Persamaan DFT adalah

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad 0 \leq k \leq N - 1 \tag{2.11}$$

$$W_N = e^{-j\frac{2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j \sin\left(\frac{2\pi}{N}\right)$$

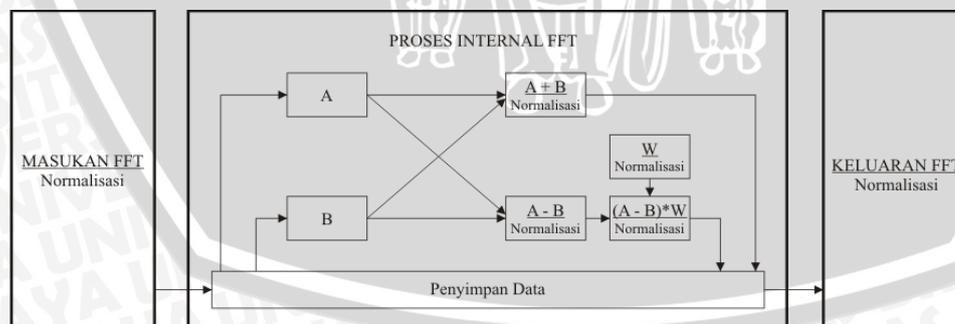
Pada persamaan (2.11),  $X[k]$  merupakan sinyal diskrit ke- $k$  dalam kawasan frekuensi,  $x[n]$  merupakan sinyal diskrit ke- $n$  dalam kawasan waktu,  $N$  merupakan jumlah titik DFT dan  $W_N$  adalah faktor perputaran (*twiddle*). Pada sistem OFDM,  $X[k]$  merupakan simbol QAM berupa bilangan kompleks yang merupakan perwakilan dari *bit*-

*bit* informasi sedangkan  $x[n]$  merupakan simbol OFDM berupa bilangan kompleks yang akan dikirimkan melalui kanal transmisi.

DFT mempunyai kelemahan yaitu memiliki beban komputasi yang cukup tinggi sehingga membutuhkan waktu yang cukup lama dalam proses komputasi dan boros dalam pembuatan perangkat keras. Oleh karena itulah digunakan algoritma FFT sebagai algoritma cepat dari DFT yang proses komputasinya menghasilkan hasil yang sama dengan proses DFT. Kelebihan FFT adalah mempunyai beban komputasi yang lebih rendah dari pada DFT sehingga lebih cepat dalam proses komputasi dibandingkan dengan DFT dan hemat dalam pembuatan perangkat keras. (Chidambaram, 2005).

IFFT digunakan pada bagian pengirim OFDM sedangkan FFT digunakan pada bagian penerima OFDM. IFFT berfungsi untuk mengubah sinyal diskrit dari kawasan frekuensi ke kawasan waktu, sedangkan FFT merupakan proses kebalikan dari IFFT. IFFT merupakan pengganti *modulator analog* sedangkan FFT merupakan pengganti *demodulator analog* pada sistem OFDM. Pada OFDM, sinyal kompleks keluaran dari pemetaan kemudian diaplikasikan ke dalam IFFT. Keluaran dari IFFT merupakan sinyal yang saling *orthogonal* yang dikenal sebagai simbol OFDM yang nantinya akan ditransmisikan pada kanal.

Gambar 2.4 menunjukkan diagram blok FFT. Bagian pertama adalah masukan FFT, pada masukan ini dilakukan normalisasi sekaligus pengaturan panjang kata. Setelah itu dilakukan proses internal yaitu proses komputasi yang dilakukan pada setiap tingkatan FFT, sehingga apabila semua titik pada satu tingkatan sudah diproses maka komputasi akan berlanjut ke tingkatan berikutnya.



**Gambar 2.4 Diagram blok FFT**

(Sumber: Setiyanto dan Mulyana, 2008)

Penyimpanan data dimaksudkan untuk menyimpan nilai titik-titik FFT yang telah diproses pada setiap tingkatan untuk menjadi masukan pada tingkatan berikutnya. Penentuan indeks untuk A dan B yang merupakan masukan kupu-kupu *Gentlemen-Sande*

pada setiap tingkatan FFT. Algoritma ini juga dapat digunakan untuk FFT berbagai titik. Keluaran tingkatan terakhir merupakan hasil akhir proses komputasi FFT. Kemudian setelah itu dilakukan pengaturan indeks keluaran FFT agar urutan indeks cuplikannya dapat sesuai.

## 2.4 FFT DIF Radix-2

FFT *radix-2* merupakan salah satu ragam FFT yang biasa digunakan. Kata *radix-2* menunjukkan setiap tingkatan FFT dibagi menjadi dua bagian. FFT *radix-2* ini dapat dirancang untuk desimasi dalam waktu DIT (*Decimation In Time*,) atau untuk desimasi dalam frekuensi DIF (*Decimation In Frequency*). FFT DIF *radix-2* lebih sering digunakan karena memiliki keunggulan dibandingkan dengan FFT DIT *radix-2*. Keunggulan tersebut diantaranya adalah lebih sederhana dalam implementasi perangkat kerasnya. Pada skripsi ini hanya akan dibahas mengenai rancangan dan simulasi FFT DIF *radix-2*.

Persamaan FFT DIF *radix-2* dibuat berdasarkan persamaan DFT yang dikelompokkan menjadi dua bagian yaitu bagian dengan indeks genap dan indeks ganjil (Chu dan George, 2000). Proses pengubahan persamaan DFT menjadi FFT DIF *radix-2* didasarkan atas sifat khusus faktor perputaran ( $W_N$ ) yang ada pada persamaan (2.8). Ada dua sifat khusus yang dimiliki oleh faktor perputaran yaitu

$$W_N^2 = e^{-j\frac{2\pi}{N}2} = e^{-j\frac{2\pi}{N/2}} \quad (2.12)$$

$$W_N^{k+\frac{N}{2}} = e^{-j\frac{2\pi}{N}(k+\frac{N}{2})} = e^{-j\frac{2\pi}{N}k} \cdot e^{-j\frac{2\pi}{N}(\frac{N}{2})} = e^{-j\frac{2\pi}{N}k} \cdot e^{-j\pi} = -e^{-j\frac{2\pi}{N}k} = -W_N^k \quad (2.13)$$

Proses pengubahan persamaan DFT menjadi persamaan FFT DIF *radix-2* adalah

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n]W_N^{kn} = \underbrace{\sum_{n=0}^{N/2-1} x[n]W_N^{kn}}_{\text{Genap}} + \underbrace{\sum_{n=N/2}^{N-1} x[n]W_N^{kn}}_{\text{Ganjil}} \\ &= \left\{ \sum_{n=0}^{N/2-1} x[n]W_N^{2kn} \right\} + \left\{ \sum_{n=N/2}^{N-1} x[n]W_N^{(2k+1)n} \right\} = \left\{ \sum_{n=0}^{N/2-1} x[n]W_{N/2}^{kn} \right\} + \left\{ \sum_{n=N/2}^{N-1} x[n]W_N^n W_{N/2}^{kn} \right\} \\ &= \left\{ \sum_{n=0}^{N/2-1} x[n]W_{N/2}^{kn} + \sum_{n=N/2}^{N-1} x[n]W_{N/2}^{kn} \right\} + \left\{ \sum_{n=0}^{N/2-1} x[n]W_N^n W_{N/2}^{kn} + \sum_{n=N/2}^{N-1} x[n]W_N^n W_{N/2}^{kn} \right\} \\ &= \left\{ \sum_{n=0}^{N/2-1} x[n]W_{N/2}^{kn} + \sum_{n=0}^{N/2-1} x[n + \frac{N}{2}]W_{N/2}^{k(n+\frac{N}{2})} \right\} \end{aligned}$$

$$\begin{aligned}
& + \left\{ \sum_{n=0}^{\frac{N-1}{2}} x[n] W_N^n W_{\frac{N}{2}}^{kn} + \sum_{n=0}^{\frac{N-1}{2}} x[n + \frac{N}{2}] W_N^{(n+\frac{N}{2})} W_{\frac{N}{2}}^{k(n+\frac{N}{2})} \right\} \\
& = \left\{ \sum_{n=0}^{\frac{N-1}{2}} x[n] W_{\frac{N}{2}}^{kn} + \sum_{n=0}^{\frac{N-1}{2}} x[n + \frac{N}{2}] W_{\frac{N}{2}}^{kn} \right\} \\
& + \left\{ \sum_{n=0}^{\frac{N-1}{2}} x[n] W_N^n W_{\frac{N}{2}}^{kn} - \sum_{n=0}^{\frac{N-1}{2}} x[n + \frac{N}{2}] W_N^n W_{\frac{N}{2}}^{kn} \right\} \\
X[k] & = \left\{ \sum_{n=0}^{\frac{N-1}{2}} \left( x[n] + x[n + \frac{N}{2}] \right) W_{\frac{N}{2}}^{kn} \right\} + \left\{ \sum_{n=0}^{\frac{N-1}{2}} \left( x[n] - x[n + \frac{N}{2}] \right) W_N^n W_{\frac{N}{2}}^{kn} \right\} \quad (2.14)
\end{aligned}$$

Pada langkah-langkah yang berakhir pada persamaan (2.14) dapat dilihat langkah pertama dalam perubahan persamaan DFT menjadi persamaan FFT DIF *radix-2* adalah mengelompokkan suku-suku pada persamaan DFT menjadi dua kelompok yaitu kelompok berindeks genap dan ganjil. Manipulasi matematis menghasilkan persamaan akhir yaitu persamaan (2.14) yang merupakan bentuk persamaan FFT DIF *radix-2*.  $X[k]$  merupakan cuplikan ke- $k$  pada kawasan frekuensi,  $x[n]$  merupakan cuplikan ke- $n$  pada kawasan waktu dengan  $N$  adalah jumlah titik FFT dan  $W_N$  adalah faktor perputaran (Chu dan George, 2000).

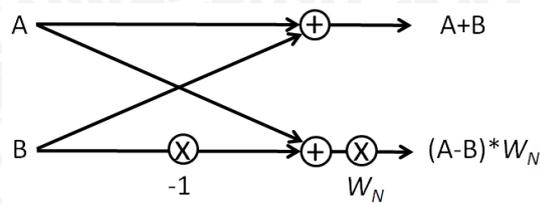
Persamaan (2.13) tersebut dapat disederhanakan lagi menjadi persamaan FFT DIF *radix-2* yang terdiri dari dua bagian DFT (Chu dan George, 2000). Hasil penyederhanaan tersebut adalah

$$\begin{aligned}
X[k] & = \left\{ \sum_{n=0}^{\frac{N-1}{2}} \left( x[n] + x[n + \frac{N}{2}] \right) W_{\frac{N}{2}}^{kn} \right\} + \left\{ \sum_{n=0}^{\frac{N-1}{2}} \left( x[n] - x[n + \frac{N}{2}] \right) W_N^n W_{\frac{N}{2}}^{kn} \right\} \\
X[k] & = DFT_{\frac{N}{2}} \left\{ x[n] + x[n + \frac{N}{2}] \right\} + DFT_{\frac{N}{2}} \left\{ W_N^n \left( x[n] - x[n + \frac{N}{2}] \right) \right\} \quad (2.15)
\end{aligned}$$

## 2.5 Diagram Kupu-kupu *Gentlemen-Sande*

Berdasarkan persamaan (2.14) tampak bahwa algoritma FFT DIF *radix-2*. Dari persamaan tersebut dapat dibuat suatu diagram yang menggambarkan operasi dasar dari

FFT DIF *radix*-2. Karena bentuknya, diagram tersebut dinamakan diagram kupu-kupu (*butterfly*) *Gentlemen-Sande* seperti terlihat pada Gambar 2.5.



**Gambar 2.5** Diagram kupu-kupu *Gentlemen-Sande*

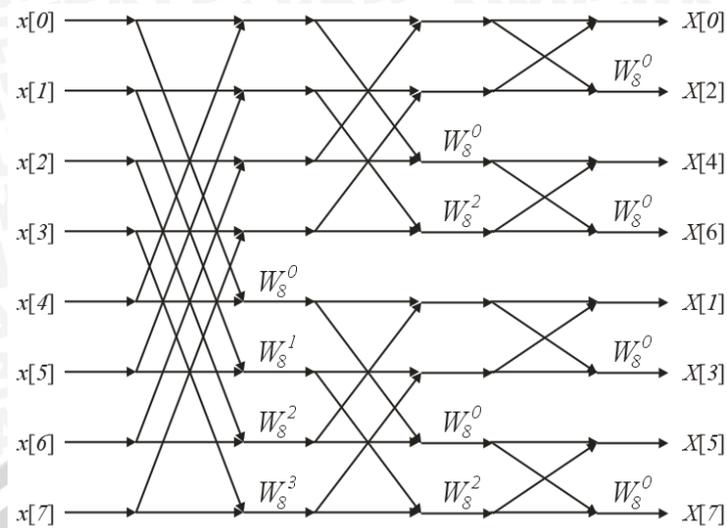
(Sumber: Chu dan George, 2000; Walters, 2007)

Masukan kupu-kupu *Gentlemen-Sande* adalah dua titik yaitu A dan B. A merupakan  $x[n]$  sedangkan B merupakan  $x[n + \frac{N}{2}]$  seperti pada persamaan (2.14).

Keluaran pertamanya adalah penjumlahan A dan B sedangkan keluaran kedua adalah hasil penjumlahan antara A dan  $-B$  yang selanjutnya dikalikan dengan  $W_N$  atau faktor perputaran.

Pada persamaan FFT DIF *radix*-2 dikenal istilah komputasi di tempat (*in place*). Komputasi di tempat adalah komputasi yang memungkinkan suatu pola perhitungan digunakan secara berulang-ulang. Kupu-kupu *Gentlemen-Sande* ini merupakan pola perhitungan yang dapat digunakan berulang-ulang pada proses komputasi FFT DIF *radix*-2. Hal ini membuat kupu-kupu ini menjadi elemen dasar komputasi yang sangat membantu dalam penyederhanaan pembuatan perangkat keras FFT DIF *radix*-2.

Gambar 2.6 menunjukkan diagram alir FFT DIF *radix*-2 untuk  $N=8$  titik, sehingga jumlah tingkatannya adalah 3. Masukannya berupa cuplikan-cuplikan ke- $n$  untuk  $n=0, 1, 2, \dots, (N-1)$  dalam kawasan waktu yaitu  $x[n]$  sedangkan keluarannya cuplikan diskrit ke- $k$  dalam kawasan frekuensi yaitu  $X[k]$ .



**Gambar 2.6** Diagram alir FFT DIF radix-2 8 Titik  
(Sumber: Chu dan George, 2000; Walters, 2007)

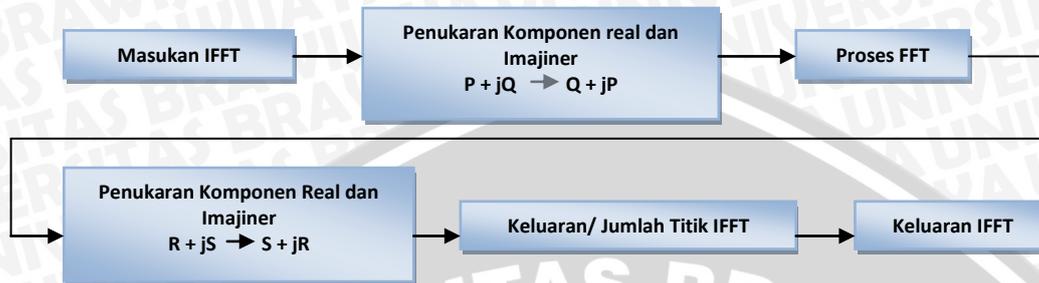
Gambar 2.6 menunjukkan susunan cuplikan pada masukannya beraturan sesuai dengan urutan pencuplikannya sedangkan susunan cuplikan pada keluarannya tidak sama dengan urutan cuplikan pada masukannya. Hal ini dikarenakan pada algoritma DIF bagian yang merupakan kawasan frekuensi atau keluaran dari FFT ini mempunyai urutan yang tidak sesuai dengan urutan pada masukannya. Tabel 2.1 menunjukkan indeks masukan dan keluaran FFT DIF radix-2 untuk  $N=8$  titik.

**Tabel 2.1** Indeks Masukan dan Keluaran FFT DIF Radix-2 8 Titik

Masukan ( $x[n]$ )			Keluaran ( $X[k]$ )		
Indeks Masukan	Bit	Indeks Cuplikan	Indeks Keluaran	Bit	Indeks Cuplikan
0	000	0	0	000	0
1	001	1	1	100	4
2	010	2	2	010	2
3	011	3	3	110	6
4	100	4	4	001	1
5	101	5	5	101	5
6	110	6	6	011	3
7	111	7	7	111	7

## 2.6 IFFT DIF *Radix-2*

IFFT DIF *radix-2* merupakan proses kebalikan dari FFT DIF *radix-2*. Modul IFFT dapat dibuat menggunakan modul FFT yang ditambahkan beberapa bagian tertentu. Diagram blok IFFT dapat dilihat pada Gambar 2.7.



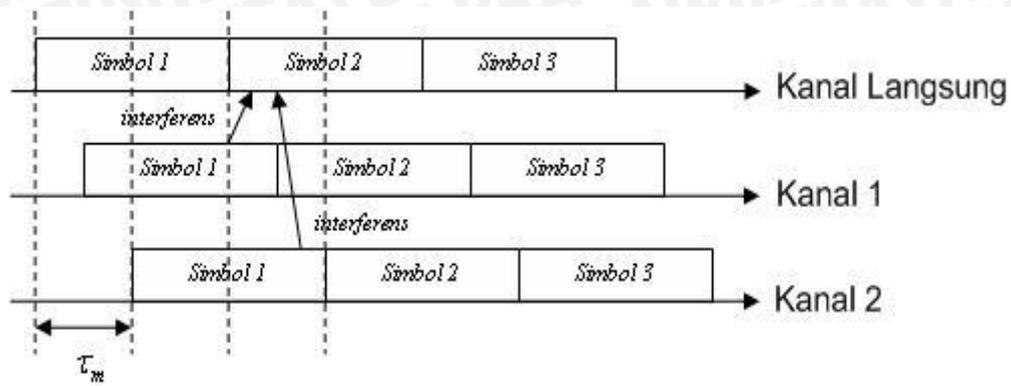
Gambar 2.7 Diagram blok IFFT  
(Sumber: Kristensen, 2005)

Proses pertama adalah menukar komponen *real* dan *imajiner* setiap masukan IFFT, selanjutnya dilakukan proses FFT terhadap masukan IFFT yang telah ditukar komponen *real* dan *imajiner* tersebut. Setelah itu dilakukan penukaran komponen *real* dan *imajiner* terhadap keluaran FFT, selanjutnya nilai setiap keluaran tersebut dibagi dengan jumlah titik IFFT/FFT.

## 2.7 Delay Spread

Ketika sinyal kirim melewati kanal transmisi maka *delay spread* (sebaran tunda) karakteristik kanal transmisi akan mempengaruhi sinyal yang akan tiba di penerima. *Delay spread* adalah salah satu faktor yang mempengaruhi karakteristik kanal transmisi. Panjang kanal dan perambatan gelombang yang bermacam-macam akibat pantulan seperti gedung, gunung, atau benda lain akan mengakibatkan sinyal melewati banyak kanal dengan berbagai variasi *delay*. *Delay* maksimum kanal adalah waktu terlama yang terjadi pada karakteristik kanal transmisi.

Waktu tempuh sinyal pantul yang tiba di penerima akan lebih lama dari waktu tempuh sinyal utama, karena jarak tempuh kanal sinyal pantul yang lebih panjang akibat pengaruh *delay*. *Delay* kanal akan menyebabkan sinyal yang tiba di penerima mengalami perubahan terhadap sinyal kirim. Hal ini disebabkan antar simbol OFDM saling bertumbukkan akibat pengaruh *delay*.



**Gambar 2.8 Delay Spread pada kanal-jamak**  
(Sumber: Faisal Muhammad, 2009:20)

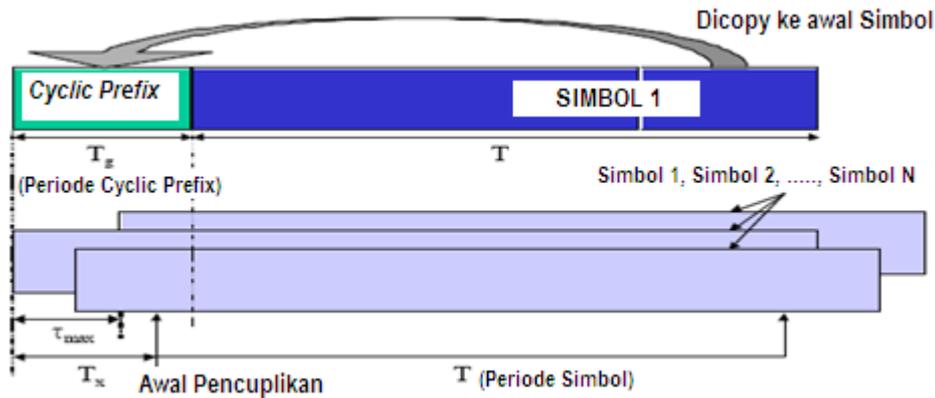
Simbol-simbol OFDM dikirim melalui kanal jamak dengan *delay* maksimum kanal  $\tau_m$ . Simbol-2 pada kanal langsung diganggu oleh simbol-1 pada kanal -1 dan kanal-2 sehingga terjadi interferensi antar simbol (ISI). Masalah ISI akan mengakibatkan sinyal yang tiba di penerima mengalami perubahan dalam kuat sinyal. Perubahan ini akan meningkatkan jumlah *bit* salah yang terjadi di penerima.

## 2.8 Interval Pengaman dan *Cyclic Prefix*

Salah satu keuntungan pada sistem OFDM adalah kemampuan dalam mengatasi interferensi antar simbol (ISI) akibat adanya pengaruh kanal jamak dan *delay spread*.

Teknik yang digunakan dalam mengatasi efek ISI adalah dengan menambahkan interval pengaman GI (*Guard Interval*) antara simbol OFDM. Masalah ISI terjadi karena simbol-simbol pada tiap kanal saling mengganggu terhadap simbol yang lain. ISI ini akan mengakibatkan jumlah *bit* salah yang terjadi pada penerima akan semakin meningkat. Masalah ISI yang timbul tidak dapat dihilangkan keseluruhan namun hanya dapat diminimalkan.

Interval pengaman yang dipilih adalah beberapa data simbol terakhir dari satu keseluruhan simbol OFDM agar ortogonalitas *subcarrier* OFDM tetap terjamin. Interval pengaman yang digunakan yaitu *Cyclic Prefix*. Penggunaan *Cyclic Prefix* dapat dilihat pada Gambar 2.9.



**Gambar 2.9 Penggunaan Cyclic Prefix**  
(Sumber: Faisal Muhammad, 2009:18)

$T_G$  adalah periode interval pengaman.  $\tau_{MAX}$  adalah besarnya *delay* kanal jamak. Panjang *Cyclic Prefix* diatur setidaknya sebesar maksimum *delay* kanal jamak. Pemilihan ukuran interval pengaman adalah sangat penting dalam mengatasi masalah ISI. Hal ini disesuaikan dengan karakteristik kanal. Penambahan interval pengaman akan membutuhkan *bandwidth* tambahan sebagai wujud kompensasi dari penggunaan interval pengaman pada simbol OFDM. Semakin besar ukuran interval pengaman yang digunakan maka semakin kecil pengaruh interferensi namun akan semakin lebar *bandwidth* yang dibutuhkan sehingga berakibat penggunaan kapasitas kanal akan semakin berkurang.

Interval pengaman *Cyclic Prefix* adalah interval pengaman yang berisi data yang berasal dari duplikat beberapa data terakhir simbol OFDM. Ortogonalitas antar *subcarrier* tetap terjaga dengan penyisipan *Cyclic Prefix* pada awal simbol OFDM. *Cyclic Prefix* yang biasa digunakan adalah 1/4, 1/8, 1/16, dan 1/32 dari titik IFFT/FFT. Penggunaan ukuran interval pengaman *Cyclic Prefix* yang berlainan akan menghasilkan kinerja sistem OFDM yang berbeda-beda.

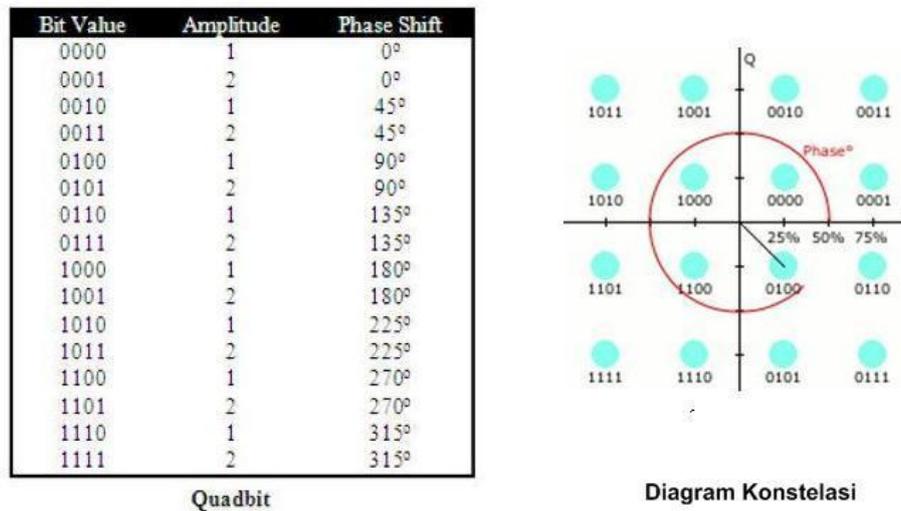
## 2.9 16-QAM (*Quadrature Amplitudo Modulation*)

QAM merupakan teknik yang mengkombinasikan antara teknik *Amplitude Shift Keying* (ASK) dan teknik *Phase Shift Keying* (PSK). Amplitudo dan fasa sinyal *carrier* pada QAM berubah terhadap perubahan amplitudo sinyal informasi, sehingga sinyal direpresentasikan dalam besaran amplitudo dan pergeseran fasa.

Dikatakan *quadrature* karena jumlah minimum *bit* pada tiap simbolnya adalah 2 *bit*. Sehingga pada QAM dikenal teknik 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, dan 256-QAM tergantung banyaknya jumlah *bit* dalam satu simbol. (K. Fazal

and S. Kaiser, 2008 : 222).

Pada skripsi ini dibahas teknik 16-QAM yaitu modulasi sinyal digital dengan penggunaan spektrum frekuensi cukup tinggi dan tingkat kehandalan yang lebih baik. Untuk 16-QAM terdapat 4 *bit* data informasi setiap simbolnya, sehingga disebut juga dengan *quadbit*. Gambar 2.10 adalah konstelasi sinyal modulasi 16-QAM.



**Gambar 2.10** Konstelasi Sinyal 16-QAM  
(Sumber : Behrouz A. Forouzan, 2000)

Gambar 2.10 menjelaskan hubungan antara amplitudo dan fasa dengan *bit* pada 16-QAM. Gambar 2.10 juga disebut dengan konstelasi yang menunjukkan hubungan yang sama dengan mengilustrasikan amplitudo dan fasanya. (Behrouz A. Forouzan, 2000).

## 2.10 AWGN (*Additive White Gaussian Noise*)

Dalam sistem komunikasi, sinyal yang diterima kemungkinan akan berbeda dengan sinyal yang dikirimkan, hal ini karena adanya berbagai gangguan yang terjadi selama proses transmisi. Pada sinyal digital, gangguan akan mengakibatkan terjadinya kesalahan pada sinyal informasi yang diterima. Gangguan-gangguan yang biasa terjadi adalah attenuasi, distorsi, *delay*, dan derau. Derau merupakan gangguan berupa sinyal-sinyal yang tidak diinginkan yang selalu muncul yang dapat menyebabkan distorsi (kerusakan) pada sinyal informasi selama proses transmisi sinyal informasi dari pengirim menuju penerima.

Pada komunikasi digital seperti sistem OFDM, kesalahan pada sinyal informasi dapat menyebabkan terjadinya kesalahan pada *bit* informasi. Hal ini tentu saja sangat merugikan dan membuat informasi yang diterima pada bagian penerima tidak sama dengan informasi yang dikirim pada bagian pengirim.

Derau Putih *Gaussian* Aditif AWGN (*Additive White Gaussian Noise*) merupakan salah satu bentuk pemodelan derau dalam suatu kanal transmisi. AWGN yang dikenal sebagai derau *Gaussian* merupakan derau yang disebabkan oleh adanya pergerakan acak elektron bebas pada rangkaian. AWGN mempunyai beberapa sifat yaitu aditif (*additive*) yang berarti derau ini akan dijumlahkan dengan sinyal informasi, putih (*white*) yang berarti derau ini memiliki rapat daya yang konstan serta tidak tergantung pada frekuensi operasi sistem dan (*Gaussian*) yang berarti tegangan derau terdistribusi secara normal atau dikenal sebagai distribusi *Gaussian*. Persamaan probabilitas derau AWGN yaitu:

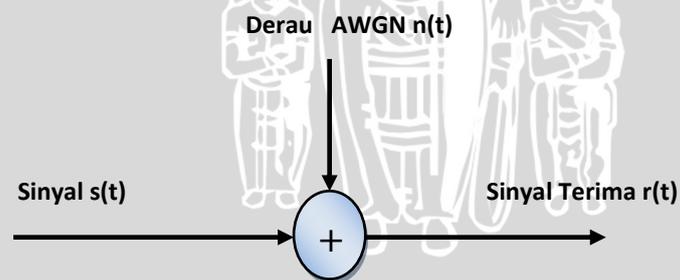
$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right] \quad (2.16)$$

Karakteristik utama dari AWGN adalah tegangan derau terdistribusi secara normal dengan rata-rata 0 dan varians  $\sigma^2$ .

SNR (*Signal to Noise Ratio*,) merupakan perbandingan daya sinyal dengan daya derau. SNR pada AWGN biasanya dinyatakan dalam satuan dB (*decibel*). Persamaan SNR yang dinyatakan dengan satuan dB adalah:

$$SNR (dB) = 10^{10} \log \frac{P_{\text{sin yal}}}{P_{\text{derau}}} \quad (2.17)$$

Pada persamaan (2.17)  $P_{\text{sin yal}}$  merupakan daya sinyal informasi dan  $P_{\text{derau}}$  merupakan daya derau.



**Gambar 2.11** Pemodelan AWGN pada kanal transmisi  
(Sumber: Faisal Muhammad, 2009:22)

Pemodelan AWGN pada kanal transmisi dapat dilihat pada Gambar 2.11. Berdasarkan gambar tersebut dapat diketahui persamaan sinyal informasi yang dipengaruhi oleh AWGN yaitu

$$r(t) = s(t) + n(t) \quad (2.18)$$

Pada persamaan (2.18)  $s(t)$  merupakan sinyal informasi,  $n(t)$  merupakan AWGN dan  $r(t)$  merupakan sinyal informasi yang dipengaruhi oleh AWGN. Pada sistem OFDM,  $s(t)$  adalah simbol OFDM yang merupakan bilangan kompleks yang terdiri dari bagian *real* dan *imajiner*. Maka sinyal AWGN yaitu  $n(t)$  dimodelkan mempunyai bagian *real* dan *imajiner*. Sehingga bagian *real* sinyal AWGN akan mempengaruhi bagian *real* sinyal informasi dan bagian *imajiner* sinyal AWGN akan mempengaruhi bagian *imajiner* sinyal informasi. Hal tersebut menyebabkan sinyal informasi yang dipengaruhi AWGN yaitu  $r(t)$  merupakan bilangan kompleks. Persamaan sinyal informasi yang dipengaruhi oleh AWGN pada sistem OFDM adalah

$$\begin{aligned} r(t) &= s(t) + n(t) = (s_{real} + j s_{imajiner}) + (n_{real} + j n_{imajiner}) \\ r(t) &= (s_{real} + n_{real}) + j (s_{imajiner} + n_{imajiner}) = r_{real} + j r_{imajiner} \end{aligned} \quad (2.19)$$

Persamaan daya AWGN atau  $p(t)$  adalah

$$p(t) = \frac{v^2(t)}{R} \quad (2.20)$$

Biasanya besar  $R$  dinyatakan dalam  $1 \Omega$  sehingga persamaan (2.20) dapat disederhanakan menjadi

$$p(t) = v^2(t) \quad \text{apabila } R = 1 \Omega \quad (2.21)$$

Persamaan daya rata-rata suatu sinyal dalam durasi waktu tertentu adalah sebagai berikut

$$P_{rata-rata} = \frac{\sum_{t=0}^T v^2(t)}{T} \quad (2.22)$$

Persamaan (2.22) ini sama dengan bentuk hubungan rata-rata (*mean*) dan *varians* ( $\sigma^2$ ) apabila nilai rata-rata = 0,  $x_i$  merupakan data ke- $i$  dan  $\bar{x}$  merupakan rata-rata. Hubungan rata-rata dan *varians* adalah

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N} \quad (2.23)$$

Berdasarkan persamaan (2.22) dan (2.23) dapat diketahui hubungan antara daya AWGN dan *varians* yaitu daya rata-rata AWGN merupakan *varians* pada AWGN yang memiliki karakteristik rata-rata = 0. Adapun hubungan antara standar deviasi dengan *varians* adalah standar deviasi merupakan akar dari *varians* yaitu

$$\text{Standar deviasi} = \sqrt{\sigma^2} = \sigma \quad (2.24)$$

Berdasarkan persamaan (2.17) dapat diketahui hubungan antara daya rata-rata AWGN ( $P_{derau}$ ) dengan SNR (dB) dan daya rata-rata sinyal ( $P_{sinyal}$ ) yaitu

$$P_{derau} = \frac{P_{sinyal}}{10^{\left(\frac{SNR (dB)}{10}\right)}} \quad (2.25)$$

### 2.11 Fungsi Korelasi Silang ( *Cross Correlation Function, CCF* )

Fungsi korelasi pada sistem telekomunikasi digunakan untuk membandingkan satu sinyal dengan sinyal lainnya untuk mencari keceratan atau keterkaitan hubungan antar kedua atau lebih dari variabel yang dibandingkan atau dikorelasikan. Salah satunya yaitu untuk proses pencarian pola simbol yang sama pada komunikasi dengan sinyal periodik, maka perlu digunakan sebuah fungsi korelasi.

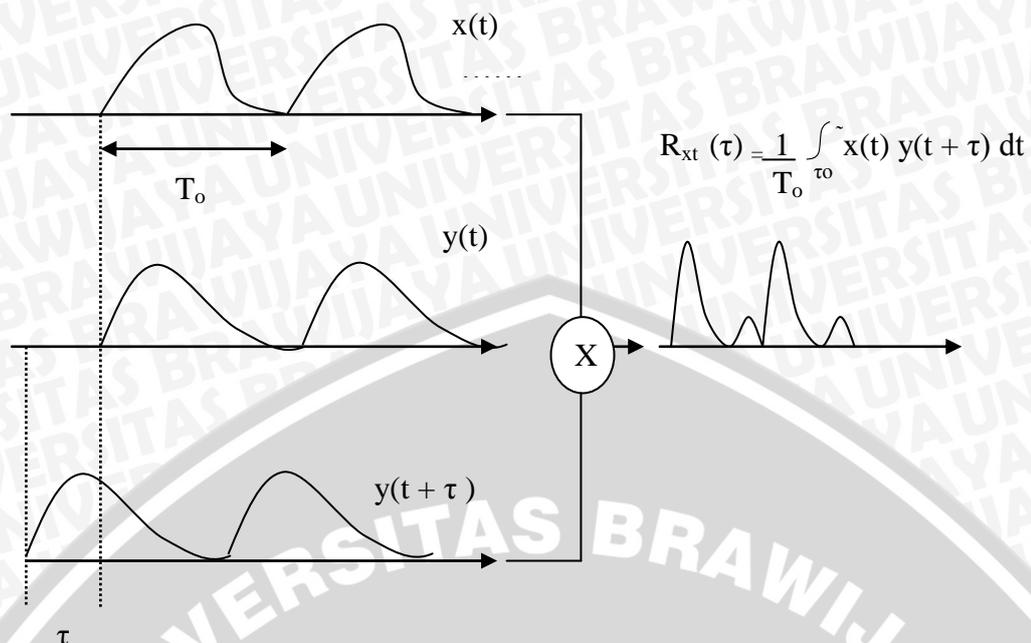
Fungsi korelasi ada dua macam yaitu fungsi korelasi diri sendiri ACF (*Auto Correlation Function*) dan fungsi korelasi silang CCF (*Cross Correlation Function*). Dalam fungsi korelasi penentuan hasil korelasi antar variabel ditentukan oleh besarnya nilai koefisien korelasi ( $r$ ). Jika nilai koefisien korelasi = 0 (nol), maka sama sekali tidak ada hubungan keterkaitan antar variabel yang dikorelasikan, jika nilai koefisien korelasi = 1 (satu), maka adanya keterkaitan yang kuat antar variabel yang dikorelasikan tersebut. Namun dalam kenyataannya, hasil koefisien korelasi bisa saja tidak mutlak 0 atau 1, sehingga untuk menentukan hasil korelasinya adalah dengan menentukan batas minimal mendekati nilai 1 untuk menyatakan bahwa adanya korelasi yang kuat antar variabel yang dikorelasikan tersebut.

Sistem simulasi pada penelitian ini menggunakan fungsi korelasi silang CCF yaitu merupakan fungsi perkalian sinyal satu dengan sinyal yang berbeda lainnya dengan pergeseran waktu ( $\tau$ ).

Dalam fungsi korelasi silang, sinyal  $x(t)$  dikalikan dengan sinyal  $y(t)$  yang telah mengalami pergeseran waktu ( $\tau$ ) maka persamaan hasil korelasi silangnya adalah:

$$R_{xt}(\tau) = \frac{1}{T_0} \int_{t_0}^{\sim} x(t) y(t + \tau) dt \quad (2.26)$$

Contoh sinyal hasil korelasi sinyal periodis dapat dilihat pada Gambar 2.12.



**Gambar 2.12 Fungsi Korelasi**  
 (Sumber: Faisal Muhammad, 2009:27)

## BAB III

### METODOLOGI PENELITIAN

Kajian dalam penelitian ini bersifat aplikatif yaitu mengkaji sebagian tahapan yang terjadi pada sistem OFDM (*Orthogonal Frequency Division Multiplexing*) yaitu mengenai pencarian jendela FFT (*Fast Fourier Transform*) pada penerima sistem OFDM yang berkaitan dengan adanya penambahan interval pengaman berupa *Cyclic Prefix* dan pengaruh besarnya derau dari kanal AWGN (*Adaptive White Gaussian Noise*).

#### 3.1 Studi Literatur

Tahapan penelitian ini meliputi penentuan parameter yang akan digunakan, jenis data, cara mendapatkan data, *instrument* yang digunakan dan proses analisis data. Parameter rancangan pada sistem ini adalah jumlah titik IFFT/FFT dengan 64 titik, modulasi digital menggunakan 16-QAM, interval *Cyclic Prefix*  $\frac{1}{4}$ , kanal transmisi dipengaruhi oleh AWGN yang memiliki derau 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt, batasan nilai koefisien korelasinya 0.8, 0.9, 1, dan interval waktu mulai dari 10 ms.

#### 3.2 Pengumpulan Data

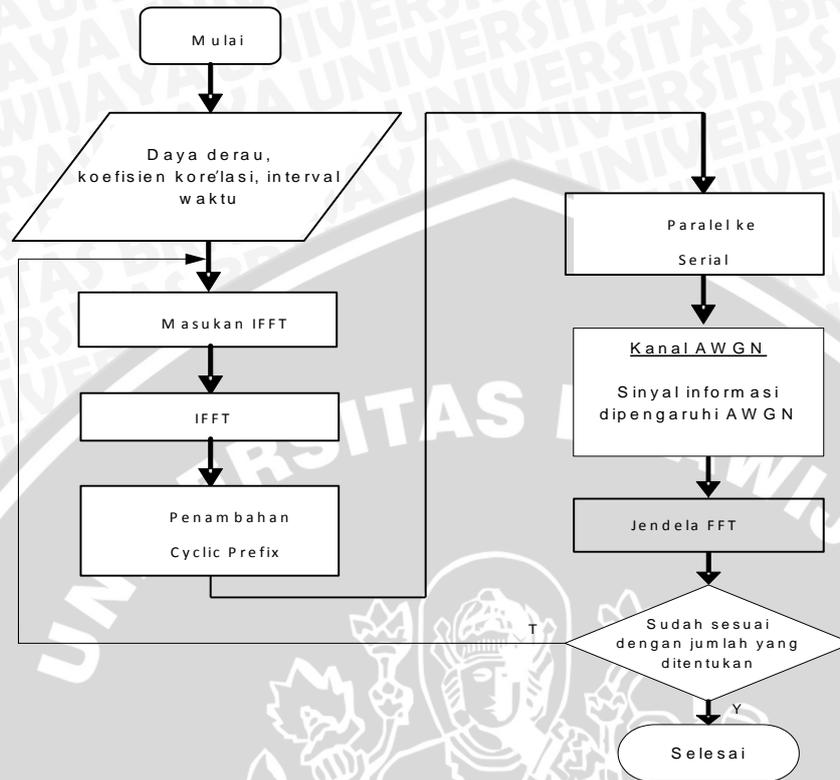
Jenis data terdiri dari data primer dan data sekunder. Data primer diperoleh dari data *real* lapangan atau hasil pengamatan secara langsung, sedangkan data sekunder diperoleh dari buku atau literatur.

##### 3.2.1 Data Sekunder

Data yang dipakai pada penelitian ini adalah data sekunder. Cara mendapatkan data diperoleh dengan penambahan *Cyclic Prefix*, pengaruh kanal AWGN, nilai *error* yang terjadi pada penentuan jendela FFT.

### 3.2.2 Simulasi Sistem

Gambar 3.1 menunjukkan diagram alir simulasi sistem pencarian jendela FFT.



**Gambar 3.1 Diagram alir Pencarian Jendela FFT**

Penjelasan diagram alir simulasi dari Gambar 3.1 adalah sebagai berikut:

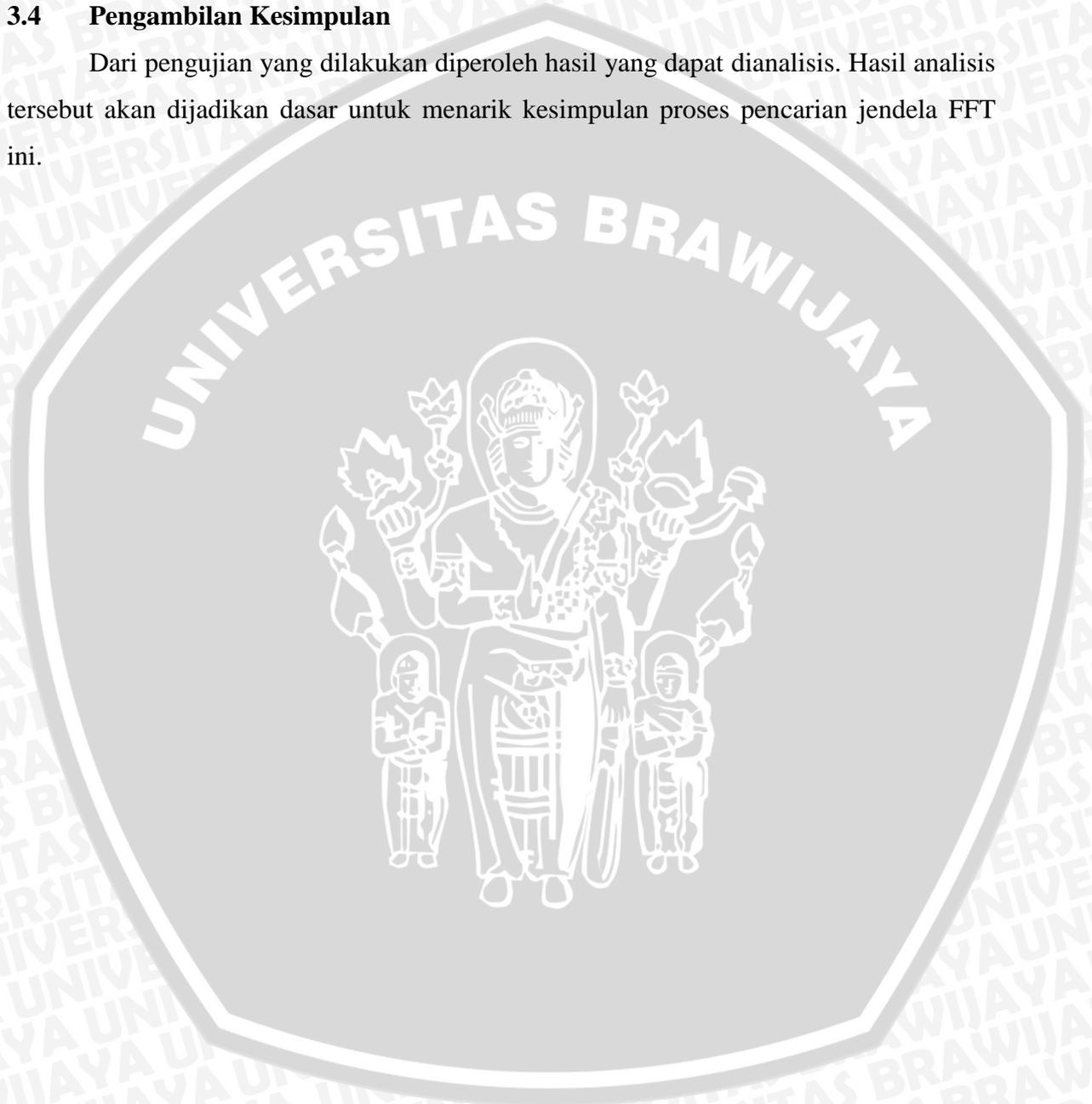
1. Inisialisasi daya derau, koefisien korelasi, interval waktu.
2. Masukan data IFFT berupa bilangan kompleks yang dibangkitkan secara acak sebanyak 64 simbol OFDM.
3. Dilakukan proses IFFT.
4. Penambahan *Cyclic Prefix* sebanyak 16 data informasi.
5. Proses konversi paralel ke serial.
6. Pengaruh AWGN pada kanal transmisi.
7. Simbol OFDM yang merupakan sinyal informasi yang diterima pada sisi penerima OFDM sebagai jendela FFT akan ditampilkan dalam simulasi sebagai hasil dari pencarian jendela FFT.
8. Jika sudah sesuai dengan dengan jumlah yang ditentukan maka program selesai, jika belum kembali ke langkah nomor 2.

### 3.3 Analisis Data

Proses analisis data diperoleh dari hasil pengamatan grafik pada sinyal keluaran dari simulasi pada penerima jendela FFT sebagai hasil dari simulasi menggunakan *software* Delphi 7.0 sebagai *instrument* simulasi.

### 3.4 Pengambilan Kesimpulan

Dari pengujian yang dilakukan diperoleh hasil yang dapat dianalisis. Hasil analisis tersebut akan dijadikan dasar untuk menarik kesimpulan proses pencarian jendela FFT ini.



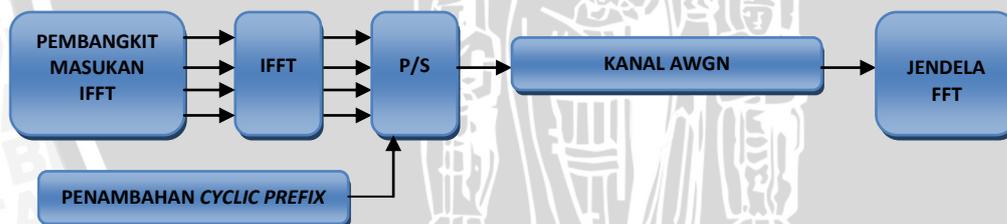
## BAB IV

### HASIL SIMULASI DAN ANALISIS

Bab ini secara khusus membahas dan menganalisis hasil perancangan dan hasil percobaan *software* simulasi. Perancangan dan pengujian *software* simulasi diperlukan untuk memastikan simulasi berjalan dengan baik sesuai dengan teori dasar dan tujuan pembuatan simulasi, sedangkan analisis hasil percobaan dilakukan agar didapatkan kesimpulan.

#### 4.1 Perancangan Simulasi Sistem

Perancangan simulasi ini dibuat menggunakan *software* Delphi 7.0. Sistem simulasi ini terdiri dari beberapa bagian utama seperti terlihat pada blok diagram sistem OFDM yang dibuat pada *software* simulasi yaitu pada Gambar 4.1. Pada blok diagram tersebut terlihat bahwa bagian-bagian Sistem OFDM pada sisi pengirim yang dibahas adalah mulai dari pembangkitan urutan bit masukan IFFT, IFFT, paralel ke serial (P/S), penambahan *Cyclic Prefix*. Kemudian kanal transmisi dipengaruhi oleh derau yaitu AWGN. Selanjutnya bagian pada penerima difokuskan untuk pencarian atau penentuan jendela FFT (*FFT Window*).



Gambar 4.1 Diagram blok sistem OFDM pada *software* simulasi

Pada Gambar 4.1 terlihat bahwa ada beberapa bagian sistem OFDM yang dihilangkan dari keseluruhan sistem OFDM baik pada bagian pengirim dan bagian penerima. Hal ini dikarenakan *software* simulasi yang dibuat memfokuskan penelitian pada pencarian jendela FFT pada bagian penerima sistem OFDM. Sehingga diasumsikan dengan jumlah sub-kanal dan titik IFFT/FFT yang sudah ditentukan, maka dilakukan proses pembangkitan data masukan IFFT yang berupa bilangan kompleks acak.

Pada pengirim dibangkitkan urutan data kompleks sebagai masukan IFFT berupa urutan data kompleks acak. Selanjutnya data tersebut masuk ke proses IFFT hingga

didapatkan simbol OFDM yang saling *orthogonal*. Sebelum masuk ke proses pengubahan dari bentuk paralel ke bentuk serial (P/S), dilakukan proses penambahan *Cyclic Prefix*. Kemudian simbol OFDM ini selanjutnya diubah dari bentuk paralel ke serial (P/S), lalu dikirimkan melalui kanal transmisi yang dipengaruhi oleh AWGN.

Pada sisi penerima dilakukan proses pencarian jendela FFT yaitu untuk mengetahui data atau sinyal informasinya saja yang dikirimkan oleh pengirim, tentunya setelah ada pengaruh dari penambahan *Cyclic Prefix* dan adanya pengaruh AWGN pada kanal transmisi menggunakan metode korelasi silang.

#### 4.2 Pengujian Software Simulasi

Pengujian *software* simulasi dilakukan untuk menguji algoritma setiap bagian pada simulasi pencarian jendela FFT untuk sistem OFDM. Hasil pada setiap bagian tersebut dibandingkan dengan hasil yang seharusnya didapatkan secara teoritis. Sedangkan pada bagian jendela FFT, hasil nilai koefisien korelasi akan dibandingkan dengan hasil nilai koefisien korelasi yang didapatkan dengan perhitungan secara manual matematis, hal ini untuk membuktikan kebenaran hasil perhitungan pada simulasi sistem dengan perhitungan secara manual.

Adapun parameter pengujian terhadap *software* simulasi ini adalah sebagai berikut :

1. Jumlah titik IFFT/FFT adalah 64 titik (diasumsikan tidak ada proses penambahan *Zeropad*).
2. Modulasi Digital adalah 16-QAM.
3. Interval *Cyclic Prefix* adalah  $\frac{1}{4}$ .
4. Kanal transmisi tidak dipengaruhi derau AWGN.
5. Batas Koefisien Korelasi adalah 1.
6. Interval waktu adalah 1000 ms (mempermudah dalam pengamatan keluaran dari grafik hasil simulasi).

Diasumsikan jumlah sub-kanal sama dengan jumlah titik IFFT/FFT karena dalam simulasi sistem ini tidak membahas masalah penambahan *zeropad* sehingga tidak ada proses penambahan *zeropad*. Pada kanal transmisi dibuat tidak adanya pengaruh derau AWGN untuk menguji apakah data yang diterima pada bagian penerima akan sama dengan data yang dikirim. Interval waktu digunakan untuk mengatur cepat lambatnya proses pada setiap bagian dalam simulasi berlangsung dimana diatur 1000 ms agar tidak terlalu cepat sehingga dapat diamati proses yang terjadi pada setiap bagian simulasi sistem.

#### 4.2.1 Tampilan *software* Simulasi

*Software* simulasi pencarian jendela FFT untuk sistem OFDM ini memiliki tampilan yang mudah digunakan. Gambar 4.2 menunjukkan tampilan awal dari simulasi sistem yang menampilkan setiap bagian sistem sesuai dengan blok diagram sistem. Maka dapat disimpulkan bahwa tampilan *software* simulasi sesuai dengan perancangan sistem.

Gambar 4.2 Tampilan *software* simulasi

#### 4.2.2 Pembangkit Masukan IFFT

Data masukan IFFT pada simulasi ini berupa pembangkitan bilangan kompleks hasil pemetaan 16-QAM yang dibangkitkan secara acak, dimana terdiri dari komponen *real* dan komponen *imajiner*. Pembangkitan data untuk masing-masing komponen *real* dan komponen *imajiner* adalah sebanyak 64 sub-kanal karena diasumsikan jumlah sub-kanal sama dengan jumlah titik IFFT/FFT yaitu 64 titik. data masukan IFFT yang dibangkitkan pada pengujian ini dapat dilihat pada Gambar 4.3.

Pembangkit Masukan IFFT	
Sub-Kanal	Real
0	3
1	-3
2	-1
3	-1
4	1
5	-1
6	-1
Sub-Kanal	Imajiner
0	1
1	1
2	3
3	-3
4	-1
5	1
6	-3

Gambar 4.3 Data masukan IFFT yang dibangkitkan

Perancangan pembangkit masukan IFFT untuk data masukan IFFT berbentuk paralel dan acak dimana ada dua bagian yaitu komponen *real* dan komponen *imajiner*. Pada pengujian simulasi sistem ini jumlah sub-kanal diasumsikan sama dengan jumlah titik IFFT/FFT yaitu 64 sub-kanal sehingga jumlah data masukan IFFT untuk *real* dan *imajiner* masing-masing ada 64 data paralel. Hal ini sesuai dengan data hasil pengujian, dapat disimpulkan bahwa data masukan IFFT yang dibangkitkan secara acak sesuai dengan perancangan sistem. Pembangkitan data masukan IFFT sama dengan membangkitkan keluaran simbol QAM dari proses pemetaan 16-QAM secara acak (*random*) terus menerus. Pada Tabel 4.1 dapat diketahui data paralel masukan awal dari IFFT OFDM.

Tabel 4.1 Hasil Pembangkitan data masukan IFFT secara acak

Sub kanal	Bit		Sandi Gray		Simbol QAM		
	Genap	Ganjil	Genap	Ganjil	I	Q	$S_k$
0	11	01	10	01	3	-1	$3 + j 1$
1	00	01	00	01	-3	-1	$-3 + j 1$
2	01	00	01	00	-1	-3	$-1 + j 3$
3	01	11	01	10	-1	3	$-1 - j 3$
4	10	10	11	11	1	1	$1 - j 1$
5	01	01	01	01	-1	-1	$-1 + j 1$
6	01	11	01	10	-1	3	$-1 - j 3$
7	11	11	10	10	3	3	$3 - j 3$
8	11	01	10	01	3	-1	$3 + j 1$

9	11	00	10	00	3	-3	$3+j3$
10	10	11	11	10	1	-3	$1-j3$
11	11	01	10	01	3	-1	$3+j1$
12	01	01	01	01	-1	-1	$-1+j1$
13	10	11	11	10	1	3	$1-j3$
14	00	11	00	10	-3	3	$-3-j3$
15	01	01	01	01	-1	-1	$-1+j1$
16	00	10	00	11	-3	1	$-3-j1$
17	11	00	10	00	3	-3	$3+j3$
18	10	11	11	10	1	3	$1-j3$
19	10	10	11	11	1	1	$1-j1$
20	11	10	10	11	3	1	$3-j1$
21	10	11	11	10	1	+3	$1-j3$
22	10	10	11	11	1	1	$1-j1$
23	00	10	00	11	-3	1	$-3-j1$
24	01	00	01	00	-1	-3	$-1+j3$
25	01	00	01	00	-1	-3	$-1+j3$
26	01	00	01	00	-1	-3	$-1+j3$
27	00	10	00	11	-3	1	$-3-j1$
28	01	11	01	10	-1	3	$-1-j3$
29	01	01	01	01	-1	-1	$-1+j1$
30	11	11	10	10	3	3	$3-j3$
31	01	10	01	11	-1	1	$-1-j1$
32	01	11	01	10	-1	3	$-1-j3$
33	11	10	10	11	3	1	$3-j1$
34	01	11	01	10	-1	3	$-1-j3$
35	10	01	11	01	1	-1	$1+j1$
36	01	00	01	00	-1	-3	$-1+j3$
37	01	01	01	01	-1	-1	$-1+j1$
38	11	10	10	11	3	1	$3-j1$
39	11	11	10	10	3	3	$3-j3$
40	11	01	10	01	3	-1	$-1+j1$
41	11	10	10	11	3	1	$3-j1$
42	10	01	11	01	1	-1	$1+j1$
43	00	01	00	01	-3	-1	$-3+j1$
44	00	01	00	01	-3	-1	$-3+j1$
45	10	11	11	10	1	3	$1-j3$
46	11	00	10	00	3	-3	$3+j3$
47	11	01	10	01	3	-1	$3-j1$

48	11	01	10	01	3	-1	$3+j1$
49	10	11	11	10	1	-3	$1+j3$
50	11	00	10	00	-3	3	$-3-j3$
51	00	00	00	00	-3	-3	$-3+j3$
52	11	10	10	11	3	1	$3-j1$
53	10	01	11	01	1	-1	$1+j1$
54	10	10	11	11	1	1	$1-j1$
55	10	01	11	01	1	-1	$1+j1$
56	10	11	11	10	1	3	$1-j3$
57	00	10	00	11	-3	1	$-3-j1$
58	00	11	00	10	-3	3	$-3-j3$
59	00	10	00	11	-3	1	$-3-j1$
60	11	01	10	01	3	-1	$3+j1$
61	10	11	11	10	1	3	$1-j3$
61	11	10	10	11	3	1	$3-j1$
63	01	01	01	01	-1	-1	$-1+j1$

Cara untuk mendapatkan data pada Tabel 4.1 diperoleh dari Tabel pemetaan 16-QAM.

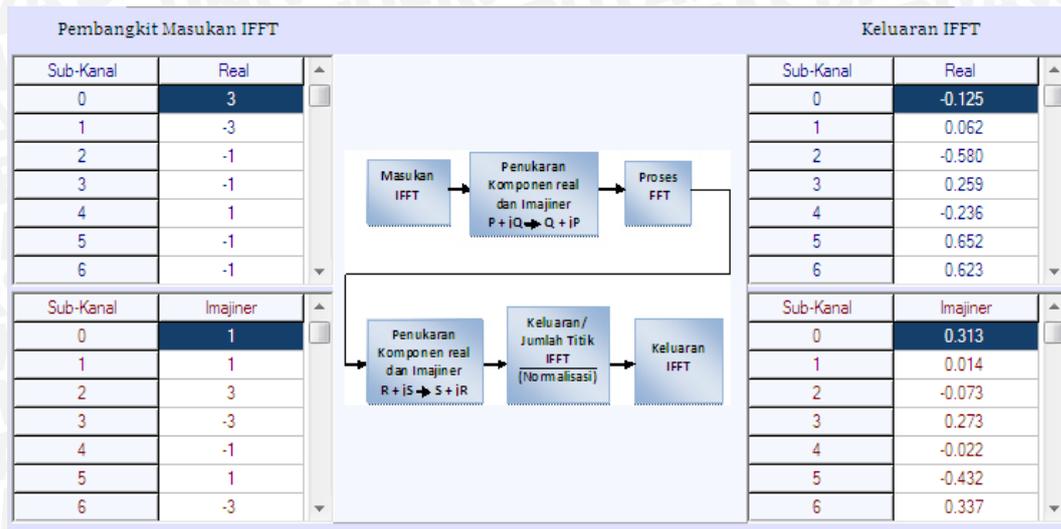
**Tabel 4.2 Pemetaan 16-QAM**  
(Sumber: Sankar Krishna, 2008)

Bit Biner		Sandi Gray		Simbol QAM
$B_0$	$B_1$	$G_0$	$G_1$	
0	0	0	0	-3
0	1	0	1	-1
1	0	1	1	1
1	1	1	0	3

Tabel 4.2 menunjukkan pemetaan 16-QAM. Sebagai contoh pada subkanal 0, sandi *Gray* genap untuk *bit* genap 11 adalah 10 dengan simbol QAM *realnya* "I" adalah 3, sandi *Gray* ganjil untuk *bit* ganjil 01 adalah 01 dengan simbol QAM *imajiner*nya "Q" adalah -1, sehingga simbol QAM kompleksnya "Sk" adalah (genap - j ganjil) atau  $(I - jQ) = (3 - j1)$ .

### 4.2.3 IFFT dan FFT

Tampilan data IFFT pada *software* simulasi ini dapat dilihat pada Gambar 4.4.



Gambar 4.4 Tampilan data IFFT

Secara lengkap data keluaran IFFT hasil pengujian *software* simulasi untuk 64 titik ini dapat dilihat pada Tabel 4.3.

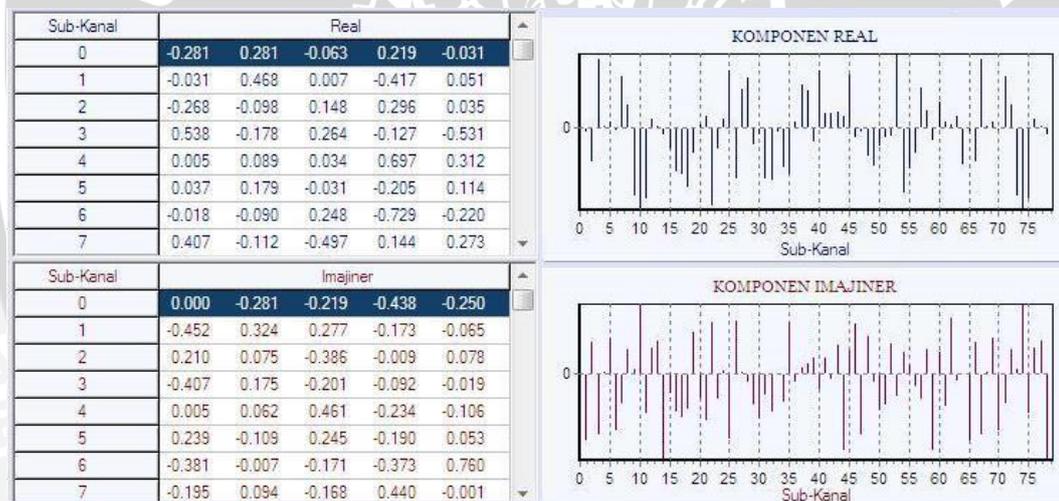
Tabel 4.3 Keluaran IFFT pada *software* simulasi

Titik IFFT	Hasil Simulasi	Titik IFFT	Hasil Simulasi
0	-0,125 + j 0,313	32	-0,250 + j 0,188
1	0,062 + j 0,014	33	0,259 + j 0,072
2	-0,580 - j 0,073	34	0,479 + j 0,653
3	0,259 + j 0,273	35	0,246 + j 0,356
4	-0,236 - j 0,022	36	-0,228 - j 0,368
5	0,652 - j 0,432	37	0,118 - j 0,040
6	0,623 + j 0,337	38	0,210 + j 0,350
7	0,170 + j 0,217	39	-0,083 + j 0,090
8	0,427 + j 0,541	40	0,073 - j 0,166
9	0,292 - j 0,309	41	-0,522 + j 0,126
10	-0,537 - j 0,301	42	0,070 + j 0,269
11	-0,041 + j 0,270	43	-0,119 + j 0,261
12	0,382 - j 0,464	44	0,008 + j 0,074
13	0,130 + j 0,211	45	-0,101 + j 0,070
14	-0,035 - j 0,096	46	-0,193 - j 0,005
15	0,290 + j 0,234	47	-0,435 + j 0,337
16	0,000 + j 0,313	48	0,375 + j 0,063
17	-0,092 + j 0,227	49	0,151 + j 0,369
18	0,165 + j 0,213	50	-0,035 - j 0,190
19	-0,284 - j 0,163	51	0,171 - j 0,348

20	-0,480 - j 0,002	52	0,193 + j 0,142
21	0,081 - j 0,190	53	0,115 - j 0,056
22	0,002 - j 0,200	54	-0,054 - j 0,133
23	-0,185 + j 0,012	55	-0,448 - j 0,707
24	-0,250 - j 0,063	56	-0,250 - j 0,063
25	0,239 + j 0,036	57	0,507 + j 0,423
26	-0,155 - j 0,176	58	-0,408 + j 0,104
27	0,145 + j 0,267	59	0,081 + j 0,188
28	-0,195 + j 0,229	60	0,055 - j 0,089
29	0,491 + j 0,245	61	0,118 - j 0,264
30	-0,170 - j 0,058	62	0,117 - j 0,194
31	-0,146 + j 0,199	63	-0,120 + j 0,014

#### 4.2.4 Penambahan *Cyclic Prefix*

Tampilan data hasil penambahan *Cyclic Prefix* pada *software* simulasi dapat dilihat pada Gambar 4.5. Sedangkan data secara lengkap dapat dilihat pada Tabel 4.4.



Gambar 4.5 Tampilan data penambahan *Cyclic Prefix*

Tabel 4.4 menunjukkan bahwa *Cyclic Prefix* ditambahkan pada sub-kanal berindeks awal dan akhir yaitu 0 sampai 15 yang diambil dari data pada sub-kanal berindeks 48 sampai 63.

Dengan penggunaan titik IFFT/FFT sebanyak 64 titik dan interval *Cyclic Prefix* sebesar  $\frac{1}{4}$ , maka akan ada  $\frac{1}{4} * 64 = 16$  data *Cyclic Prefix* yang akan ditambahkan, yaitu diambil dari 16 data terakhir dari 64 data informasi yang kemudian diduplikatkan ke bagian awal yaitu sebagai *Cyclic Prefix*, sehingga setelah ada penambahan *Cyclic Prefix*

maka sekarang data yang akan dikirim menjadi 80 data yaitu yang mempunyai indeks sub-kanal 0 sampai 79 .

Hasil percobaan terbukti bahwa hasil penambahan *Cyclic Prefix* simulasi sesuai dengan aturan secara teoritis, maka dapat disimpulkan bahwa penambahan *Cyclic Prefix* sesuai dengan perancangan sistem.

**Tabel 4.4** Data hasil penambahan *Cyclic Prefix*

Sub Kanal	Data	Sub Kanal	Data	Sub Kanal	Data
0	$0,375 + j0,063$	27	$-0,041 + j0,270$	54	$0,210 + j 0,350$
1	$0,151 + j0,369$	28	$0,382 - j0,464$	55	$-0,083 + j 0,090$
2	$-0,035 - j0,190$	29	$0,130 + j 0,211$	56	$0,073 - j 0,166$
3	$0,171 - j0,348$	30	$-0,03 - j0,096$	57	$-0,522 + j 0,126$
4	$0,193 + j0,142$	31	$0,290 + j0,234$	58	$0,070 + j 0,269$
5	$0,115 - j0,056$	32	$0,000 + j0,313$	59	$-0,119 + j 0,261$
6	$-0,054 - j0,133$	33	$-0,092 + j0,227$	60	$0,008 + j 0,074$
7	$-0,448 - j0,707$	34	$0,165 + j0,213$	61	$-0,101 + j 0,070$
8	$-0,250 - j0,063$	35	$-0,284 - j0,163$	62	$-0,193 - j 0,005$
9	$0,507 + j0,423$	36	$-0,480 - j0,002$	63	$-0,435 + j 0,337$
10	$-0,408 + j0,104$	37	$0,081 - j0,190$	64	$0,375 + j0,063$
11	$0,081 + j0,188$	38	$0,002 - j0,200$	65	$0,151 + j0,369$
12	$0,055 - j0,089$	39	$-0,185 + j0,012$	66	$-0,035 - j0,190$
13	$0,118 - j0,264$	40	$-0,250 - j0,063$	67	$0,171 - j0,348$
14	$0,117 - j0,194$	41	$0,239 + j0,036$	68	$0,193 + j0,142$
15	$-0,120 + j0,014$	42	$-0,155 - j0,176$	69	$0,115 - j0,056$
16	$-0,125 + j0,313$	43	$0,145 + j0,267$	70	$-0,054 - j0,133$
17	$0,062 + j0,014$	44	$-0,195 + j0,229$	71	$-0,448 - j0,707$
18	$-0,580 - j0,073$	45	$0,491 + j0,245$	72	$-0,250 - j0,063$
19	$0,259 + j0,273$	46	$-0,170 - j0,058$	73	$0,507 + j0,423$
20	$-0,236 - j0,022$	47	$-0,146 + j0,199$	74	$-0,408 + j0,104$
21	$0,652 - j 0,432$	48	$-0,250 + j0,188$	75	$0,081 + j0,188$
22	$0,623 + j0,337$	49	$0,259 + j0,072$	76	$0,055 - j0,089$
23	$0,170 + j0,217$	50	$0,479 + j0,653$	77	$0,118 - j0,264$
24	$0,427 + j0,541$	51	$0,246 + j0,356$	78	$0,117 - j0,194$
25	$0,292 - j 0,309$	52	$-0,228 - j0,368$	79	$-0,120 + j0,014$
26	$-0,537 - j0,301$	53	$0,118 - j0,040$		

Dengan penggunaan titik IFFT/FFT sebanyak 64 titik dan interval *Cyclic Prefix* sebesar  $\frac{1}{4}$ , maka akan ada  $\frac{1}{4} * 64 = 16$  data *Cyclic Prefix* yang akan ditambahkan, yaitu diambil dari 16 data terakhir dari 64 data informasi (sub kanal 48 sampai sub kanal 63) yang kemudian diduplikatkan ke bagian awal yaitu sebagai *Cyclic Prefix* (sub kanal 0 sampai sub kanal 15) sehingga setelah ada penambahan *Cyclic Prefix* maka sekarang data yang akan dikirim menjadi 80 data yaitu yang mempunyai indeks sub-kanal 0 sampai 79 .

#### 4.2.5 Paralel ke Serial ( P/S )

Pada bagian ini terdapat proses paralel ke serial yang mengubah data paralel hasil keluaran IFFT menjadi bentuk serial sebelum dikirimkan ke penerima melalui kanal transmisi. Tampilan data serial pada *software* simulasi pada Gambar 5.5 diperoleh dari data keluaran setelah adanya proses penambahan *Cyclic Prefix* (berjumlah 80) dimana kompoen *real* dan *imaginer* ditampilkan secara terpisah. Dari hasil pengujian tersebut terlihat bahwa keluarannya berbentuk serial. Maka dapat disimpulkan bahwa proses paralel ke serial pada bagian pengirim sesuai dengan perancangan sistem.

Komponen Real Serial																
-0.161	-0.051	0.008	0.070	-0.552	-0.659	-0.532	0.180	0.407	-0.018	0.037	0.005	0.538	-0.268	-0.031	-0.281	0.083
0.022	0.044	0.199	-0.093	0.131	0.311	-0.195	-0.315	-0.509	0.585	-0.063	-0.074	-0.134	-0.300	-0.219	-0.024	-0.072
0.411	0.084	0.119	0.108	0.114	0.445	-0.110	0.288	0.337	0.039	-0.360	-0.311	-0.022	-0.406	-0.399	-0.045	-0.126
0.397	0.298	-0.403	0.444	0.070	-0.157	-0.615	0.088	0.019	-0.198	-0.464	-0.368	-0.344	-0.161	-0.051	0.008	0.070
-0.552	-0.659	-0.532	0.180	0.407	-0.018	0.037	0.005	0.538	-0.268	-0.031	-0.281	0.205	0.226	-0.084	-0.346	-0.352
-0.024	-0.208	0.070	-0.112	-0.090	0.179	0.089	-0.178	-0.098	0.468	0.281	-0.158	-0.121	-0.150	-0.226	-0.115	-0.125
-0.252	0.003	-0.328	-0.347	-0.042	0.311	-0.453	-0.231	-0.036	0.281	-0.307	-0.013	-0.253	0.239	-0.254	-0.591	0.439
-0.195	-0.515	0.642	0.652	0.444	-0.055	-0.340	-0.546	0.031	-0.271	-0.232	0.104	0.208	-0.027	-0.047	-0.336	0.622
0.058	0.186	-0.480	-0.219	0.361	-0.295	0.044	-0.094	0.205	0.226	-0.084	-0.346	-0.352	-0.024	-0.208	0.070	-0.112
-0.090	0.179	0.089	-0.178	-0.098	0.468	0.281	-0.176	0.040	0.116	0.056	0.301	-0.136	-0.227	-0.313	-0.497	0.248
Komponen Imajiner Serial																
-0.125	-0.595	0.221	0.171	-0.264	0.475	0.023	0.164	-0.195	-0.381	0.239	0.005	-0.407	0.210	-0.452	0.000	-0.041
0.380	-0.212	0.142	-0.521	0.159	-0.168	-0.075	0.060	0.143	-0.147	0.202	-0.208	-0.249	-0.054	0.250	-0.415	0.339
0.154	-0.517	0.195	-0.034	0.105	-0.101	0.102	0.065	0.040	-0.049	0.349	-0.183	-0.005	-0.250	-0.142	-0.301	-0.205
-0.046	0.005	0.358	-0.437	0.013	-0.171	0.350	-0.310	-0.158	0.277	-0.234	-0.293	-0.250	-0.125	-0.595	0.221	0.171
-0.264	0.475	0.023	0.164	-0.195	-0.381	0.239	0.005	-0.407	0.210	-0.452	0.000	-0.016	-0.225	0.175	-0.144	-0.255
-0.431	0.102	-0.357	0.094	-0.007	-0.109	0.062	0.175	0.075	0.324	-0.281	0.539	-0.334	0.463	0.192	0.009	-0.093
-0.208	0.169	0.176	0.876	-0.043	0.208	-0.084	-0.045	-0.052	0.406	-0.028	-0.181	-0.398	-0.077	-0.126	-0.442	-0.198
-0.268	0.231	0.236	-0.193	0.321	0.233	-0.302	0.184	-0.031	-0.252	-0.035	0.320	0.029	-0.322	-0.174	-0.267	0.081
-0.569	0.419	-0.037	-0.091	0.193	-0.338	-0.062	-0.219	-0.016	-0.225	0.175	-0.144	-0.255	-0.431	0.102	-0.357	0.094
-0.007	-0.109	0.062	0.175	0.075	0.324	-0.281	-0.088	-0.508	0.468	-0.355	0.109	0.129	0.699	-0.173	-0.168	-0.171

Gambar 4.6 Tampilan data serial pada P/S di bagian pengirim

Pada bagian ini terdapat proses paralel ke serial yang mengubah data paralel hasil keluaran IFFT menjadi bentuk serial sebelum dikirimkan ke penerima melalui kanal transmisi. Data serial ini diperoleh dari data keluaran setelah adanya proses penambahan *Cyclic Prefix* (berjumlah 80 dari sub kanal 0 sampai sub kanal 79) dimana kompoen *real* dan *imaginer* ditampilkan secara terpisah.

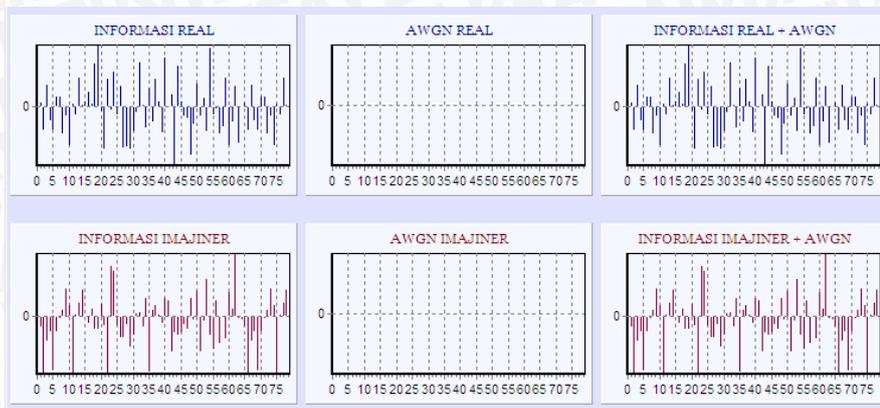
#### 4.2.6 Kanal AWGN

Pada pengujian bagian ini, karena pada parameter masukan tidak diatur adanya pengaruh derau pada kanal AWGN ( derau = 0 ) maka data yang dikirim oleh bagian pengirim tidak terpengaruh AWGN, sehingga data yang dikirimkan akan sama dengan data yang diterima setelah melewati kanal transmisi. Tampilan data informasi setelah melewati kanal transmisi dapat dilihat pada Gambar 4.7, dimana datanya terdiri dari komponen *real* dan komponen *imajiner*. Pengaturan tidak adanya pengaruh derau pada kanal AWGN untuk menguji apakah data yang diterima pada bagian penerima akan sama dengan data yang dikirim. Berdasarkan hasil pengujian ternyata data yang diterima sama dengan data yang dikirim atau dengan kata lain tidak ada perubahan. Berdasarkan penjelasan tersebut, maka dapat disimpulkan bahwa kanal AWGN sesuai dengan perancangan sistem.

Informasi + AWGN (Real)																
-0.161	-0.051	0.008	0.070	-0.552	-0.659	-0.532	0.180	0.407	-0.018	0.037	0.005	0.538	-0.268	-0.031	-0.281	0.083
0.022	0.044	0.199	-0.093	0.131	0.311	-0.195	-0.315	-0.509	0.585	-0.063	-0.074	-0.134	-0.300	-0.219	-0.024	-0.072
0.411	0.084	0.119	0.108	0.114	0.445	-0.110	0.288	0.337	0.039	-0.360	-0.311	-0.022	-0.406	-0.399	-0.045	-0.126
0.397	0.298	-0.403	0.444	0.070	-0.157	-0.615	0.088	0.019	-0.198	-0.464	-0.368	-0.344	-0.161	-0.051	0.008	0.070
-0.552	-0.659	-0.532	0.180	0.407	-0.018	0.037	0.005	0.538	-0.268	-0.031	-0.281	0.205	0.226	-0.084	-0.346	-0.352
-0.024	-0.208	0.070	-0.112	-0.090	0.179	0.089	-0.178	-0.098	0.468	0.281	-0.158	-0.121	-0.150	-0.226	-0.115	-0.125
-0.252	0.003	-0.328	-0.347	-0.042	0.311	-0.453	-0.231	-0.036	0.281	-0.307	-0.013	-0.253	0.239	-0.254	-0.591	0.439
-0.195	-0.515	0.642	0.652	0.444	-0.055	-0.340	-0.546	0.031	-0.271	-0.232	0.104	0.208	-0.027	-0.047	-0.336	0.622
0.058	0.186	-0.480	-0.219	0.361	-0.295	0.044	-0.094	0.205	0.226	-0.084	-0.346	-0.352	-0.024	-0.208	0.070	-0.112
-0.090	0.179	0.089	-0.178	-0.098	0.468	0.281	-0.176	0.040	0.116	0.056	0.301	-0.136	-0.227	-0.313	-0.497	0.248
Informasi + AWGN (Imajiner)																
-0.125	-0.595	0.221	0.171	-0.264	0.475	0.023	0.164	-0.195	-0.381	0.239	0.005	-0.407	0.210	-0.452	0.000	-0.041
0.380	-0.212	0.142	-0.521	0.159	-0.168	-0.075	0.060	0.143	-0.147	0.202	-0.208	-0.249	-0.054	0.250	-0.415	0.339
0.154	-0.517	0.195	-0.034	0.105	-0.101	0.102	0.065	0.040	-0.049	0.349	-0.183	-0.005	-0.250	-0.142	-0.301	-0.205
-0.046	0.005	0.358	-0.437	0.013	-0.171	0.350	-0.310	-0.158	0.277	-0.234	-0.293	-0.250	-0.125	-0.595	0.221	0.171
-0.264	0.475	0.023	0.164	-0.195	-0.381	0.239	0.005	-0.407	0.210	-0.452	0.000	-0.016	-0.225	0.175	-0.144	-0.255
-0.431	0.102	-0.357	0.094	-0.007	-0.109	0.062	0.175	0.075	0.324	-0.281	0.539	-0.334	0.463	0.192	0.009	-0.093
-0.208	0.169	0.176	0.876	-0.043	0.208	-0.084	-0.045	-0.052	0.406	-0.028	-0.181	-0.398	-0.077	-0.126	-0.442	-0.198
-0.268	0.231	0.236	-0.193	0.321	0.233	-0.302	0.184	-0.031	-0.252	-0.035	0.320	0.029	-0.322	-0.174	-0.267	0.081
-0.569	0.419	-0.037	-0.091	0.193	-0.338	-0.062	-0.219	-0.016	-0.225	0.175	-0.144	-0.255	-0.431	0.102	-0.357	0.094
-0.007	-0.109	0.062	0.175	0.075	0.324	-0.281	-0.088	-0.508	0.468	-0.355	0.109	0.129	0.699	-0.173	-0.168	-0.171

Gambar 4.7 Tampilan data informasi setelah terpengaruh kanal AWGN

Sinyal informasi dan sinyal AWGN yang telah dipengaruhi oleh AWGN pada pengujian ini juga dapat dilihat pada Gambar 4.8. Sinyal AWGN ini terbagi menjadi dua komponen yaitu komponen *real* dan *imajiner*. Derau pada pengujian ini adalah 0 ( tidak ada pengaruh kanal AWGN ), sehingga tidak ada sinyal AWGN yang dibangkitkan. Maka hasil sinyal informasi yang telah dipengaruhi oleh AWGN akan sama dengan sinyal informasi yang dikirim oleh bagian pengirim.



Gambar 4.8 Tampilan sinyal pada kanal AWGN

Gambar 4.8 menunjukkan bahwa sinyal informasi yang belum dan sudah dipengaruhi oleh AWGN adalah sama. Hal ini dikarenakan tidak ada sinyal AWGN yang dibangkitkan sehingga tidak mempengaruhi sinyal informasi. Berdasarkan penjelasan tersebut, maka dapat disimpulkan bahwa kanal AWGN sesuai dengan perancangan sistem.

#### 4.2.7 Jendela FFT

Pada bagian ini dilakukan 3 proses penampilan yaitu pencuplikan awal, pencarian jendela FFT, hasil jendela FFT. Pada proses pencuplikan awal, dilakukan proses pencuplikan data sebanyak 80 data masing-masing untuk komponen *real* dan komponen *imaginer* dalam waktu yang bersamaan, dengan pencuplikan tersebut diambil 16 data awal dan 16 data akhir, dimana 16 data awal tersebut merupakan *Cyclic Prefix* yang mana nantinya akan dibandingkan untuk mencari tingkat kemiripannya dengan cara proses korelasi silang. Kemudian semua pencuplikan tersebut ditampilkan. Proses yang selanjutnya yaitu proses pencarian jendela FFT dengan cara mengkorelasikan silang 16 data yang dibandingkan tersebut sehingga menghasilkan nilai koefisien korelasi secara matematis dengan batas nilai koefisien korelasi yang telah ditentukan sebelumnya.

Pada simulasi sistem ini, karena adanya pengaruh derau, maka data yang dikirim tidak akan sama dengan data yang diterima karena adanya pengaruh derau kanal AWGN, sehingga untuk menentukan tingkat korelasinya tidak dapat mutlak berdasarkan nilai koefisien korelasi 0 dan 1. Sehingga diambil batas minimal tingkat korelasinya kuat jika nilai koefisien korelasinya  $\geq 0,8$ . Pada proses pencarian, terdapat penghitung (*counter*) untuk menghitung berapa kali proses pencarian dan untuk mengetahui berapa waktu yang dibutuhkan untuk dapat menemukan jendela FFT dalam satuan bingkai (*frame*), karena

jika nilai koefisien korelasinya  $< 0,8$  maka otomatis akan melakukan proses pembandingan data selanjutnya yaitu 16 data yang digeser 1 *chip* dari pencuplikan 16 data awal. Demikian sampai didapat nilai 0,8 maka penghitung (*counter*) akan berhenti. Kemudian data yang merupakan jendela FFT yaitu 64 data setelah dikurangi *Cyclic Prefix* yang ditampilkan pada penampil ketiga yaitu pada hasil jendela FFT.

Karena dalam pengujian pengaturan parameternya tidak dipengaruhi derau pada kanal AWGN, maka nilai koefisien korelasinya adalah 1, yang artinya bahwa data yang dibandingkan mempunyai korelasi yang kuat karena data yang dikirim sama dengan data yang diterima. Tampilan data cuplikan awal dapat dilihat pada Gambar 4.9.

Informasi + AWGN (Cuplik Real)								Informasi + AWGN (Cuplik Imajiner)							
0.041	-0.124	0.325	-0.613	0.300	-0.277	-0.256	0.296	-0.429	0.006	0.089	-0.112	-0.085	0.256	0.136	-0.042
0.177	-0.222	0.095	0.287	0.375	-0.406	-0.109	0.195	-0.006	0.416	0.171	-0.256	-0.500	0.279	-0.031	0.037
0.504	0.179	-0.319	-0.254	0.101	-0.152	-0.236	-0.118	-0.300	0.111	-0.203	0.092	0.047	0.374	-0.163	-0.205
-0.158	0.095	0.164	0.087	0.313	0.032	-0.020	-0.204	0.142	-0.304	0.136	0.267	-0.031	0.084	-0.291	0.021
0.103	0.115	-0.529	0.047	0.112	-0.508	-0.186	0.092	-0.354	0.078	0.259	0.454	-0.063	0.051	-0.139	0.713
0.005	-0.021	-0.130	0.616	0.188	-0.059	0.203	-0.294	-0.219	0.315	-0.379	-0.024	-0.281	-0.165	-0.018	0.444
-0.012	-0.189	0.345	-0.327	-0.164	-0.030	0.234	0.306	0.271	0.567	0.334	-0.192	0.578	-0.264	-0.368	0.185
0.202	0.007	0.190	0.060	0.313	-0.431	0.229	-0.072	0.542	-0.137	0.778	-0.003	-0.656	0.138	-0.211	-0.453
0.655	0.200	-0.225	-0.090	0.200	0.096	0.137	0.346	-0.117	0.177	0.235	0.156	-0.063	0.255	0.472	0.008
-0.049	-0.429	-0.245	0.111	0.188	-0.406	-0.109	0.195	0.285	-0.056	0.090	0.000	-0.281	0.279	-0.031	0.037
Informasi Cuplik: 253 - 332															
Informasi + AWGN (Korelasi Real)								Informasi + AWGN (Korelasi Imajiner)							
0.655	0.200	-0.225	-0.090	0.200	0.096	0.137	0.346	-0.117	0.177	0.235	0.156	-0.063	0.255	0.472	0.008
-0.049	-0.429	-0.245	0.111	0.188	-0.406	-0.109	0.195	0.285	-0.056	0.090	0.000	-0.281	0.279	-0.031	0.037
0.041	-0.124	0.325	-0.613	0.300	-0.277	-0.256	0.296	-0.429	0.006	0.089	-0.112	-0.085	0.256	0.136	-0.042
0.177	-0.222	0.095	0.287	0.375	-0.406	-0.109	0.195	-0.006	0.416	0.171	-0.256	-0.500	0.279	-0.031	0.037

Gambar 4.9 Tampilan cuplikan awal

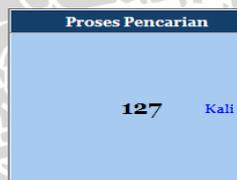
Terdapat 80 data hasil cuplikan setelah dipengaruhi oleh derau AWGN untuk masing-masing komponen *real* dan *imajiner* pada tampilan Gambar 4.9. Sedangkan cuplikan data korelasi diambil dari 16 data awal dan 16 data terakhir, dimana 16 data awal ditambahkan dari 16 data terakhir.

Untuk menguji proses korelasinya, hasil perhitungan korelasi silang yang ditampilkan pada simulasi dibandingkan dengan perhitungan secara manual matematisnya untuk menguji hasil perhitungan tersebut benar atau tidak. Tampilan proses pencarian jendela FFT dapat dilihat pada Gambar 4.10.

Cuplikan Awal					Pencarian Jendela FFT					Hasil Jendela FFT					
-0.158	-0.099	-0.084	0.250	-0.236	-0.107	0.094	0.089	0.285	0.099	-0.901	-0.119	0.084	-0.120	-0.275	-0.275
-0.313	-0.013	-0.050	0.168	0.477	-0.076	-0.096	-0.040	0.313	-0.267	-0.112	-0.144	0.020	-0.021	-0.582	-0.244
-0.397	-0.727	0.478	-0.301	-0.064	0.023	0.026	-0.267	-0.094	-0.130	0.209	-0.013	-0.038	0.182	-0.223	0.068
0.281	0.088	-0.019	0.540	-0.468	0.244	0.059	0.140	0.281	0.151	-0.117	0.028	-0.234	-0.330	0.305	0.384
-0.467	-0.296	0.348	0.315	0.317	-0.044	-0.468	-0.301	-0.152	-0.005	-0.493	-0.237	-0.253	-0.256	0.383	0.361
0.000	0.033	0.233	-0.140	-0.568	0.252	0.524	0.399	0.063	-0.143	-0.538	-0.167	0.289	-0.060	0.044	0.079
-0.353	0.089	0.200	-0.147	0.233	-0.100	-0.094	-0.534	0.657	0.395	0.472	0.597	-0.042	0.307	-0.099	0.127
0.406	0.071	-0.105	-0.331	0.299	-0.338	-0.046	0.161	-0.031	0.005	0.231	-0.047	-0.075	-0.305	0.196	0.103
-0.158	-0.099	-0.084	0.250	-0.236	-0.107	0.094	0.089	-0.285	0.099	-0.901	-0.119	0.084	-0.120	-0.275	-0.275
-0.313	-0.013	-0.050	0.168	0.477	-0.076	-0.096	-0.040	0.313	-0.267	-0.112	-0.144	0.020	-0.021	-0.582	-0.244
CP	A	B	A x B	B x B	CP	A	B	A x B	B x B						
0	-0.040	-0.040	0.002	0.002	0	-0.244	-0.244	0.060	0.060						
1	-0.096	-0.096	0.009	0.009	1	-0.582	-0.582	0.339	0.339						
2	-0.076	-0.076	0.006	0.006	2	-0.021	-0.021	0.000	0.000						
3	0.477	0.477	0.228	0.228	3	0.020	0.020	0.000	0.000						
4	0.168	0.168	0.028	0.028	4	-0.144	-0.144	0.021	0.021						
5	-0.050	-0.050	0.003	0.003	5	-0.112	-0.112	0.013	0.013						
6	-0.013	-0.013	0.000	0.000	6	-0.267	-0.267	0.071	0.071						
7	-0.313	0.000	-0.000	0.000	7	0.313	0.000	0.000	0.000						
8	0.089	0.000	0.000	0.000	8	-0.275	0.000	-0.000	0.000						
9	0.094	0.000	0.000	0.000	9	-0.275	0.000	-0.000	0.000						
Jumlah				0.562	0.562	Jumlah				1,692	1,692				
Koefisien Korelasi : Real				0.562 / 0.562	: 1	Koefisien Korelasi : Imajiner				1,692 / 1,692	: 1				

Gambar 4.10 Tampilan pencarian jendela FFT

Untuk mengetahui secara lengkap mengenai data yang akan dikorelasikan dapat dilihat pada Tabel 4.5. Seperti terlihat pada Gambar 4.11, berdasarkan satu kali pengujian, ternyata dibutuhkan proses sebanyak 127 bingkai (*frame*) untuk dapat menemukan mana yang merupakan jendela FFT yang dikirimkan, dengan kata lain ada sebanyak 127 kali proses pencuplikan dan korelasi antar kedua data yang dibandingkan tersebut tidak kuat.



Gambar 4.11 Tampilan penghitung proses pencarian jendela FFT

Tabel 4.5 Data yang akan dikorelasikan

No	Komponen Real		Komponen Imajiner	
	Data 1	Data 2	Data 1	Data 2
1	-0,040	-0,040	-0,244	-0,244
2	-0,096	-0,096	-0,582	-0,582
3	-0,076	-0,076	-0,021	-0,021
4	0,477	0,477	-0,020	-0,020
5	0,168	0,168	-0,144	-0,144
6	-0,050	-0,050	-0,112	-0,112
7	-0,013	-0,013	-0,267	-0,267
8	-0,313	-0,313	0,313	0,313

9	0,089	0,089	-0,275	-0,275
10	0,094	0,094	-0,275	-0,275
11	-0,107	-0,107	-0,120	-0,120
12	-0,236	-0,236	0,084	0,084
13	0,250	0,250	-0,119	-0,119
14	-0,084	-0,084	-0,901	-0,901
15	-0,099	-0,099	0,099	0,099
16	-0,158	-0,158	-0,285	-0,285

Perhitungan secara matematis untuk mencari nilai koefisien korelasi tersebut adalah sebagai berikut:

- Komponen *Real*

<u>Data 1</u>		<u>Data 2</u>		
-0,040	x	-0,040	=	0,002
-0,096	x	-0,096	=	0,009
-0,076	x	-0,076	=	0,006
0,477	x	0,477	=	0,228
0,168	x	0,168	=	0,028
-0,050	x	-0,050	=	0,003
-0,013	x	-0,013	=	0,000
-0,313	x	-0,313	=	0,098
0,089	x	0,089	=	0,008
0,094	x	0,094	=	0,009
-0,107	x	-0,107	=	0,011
-0,236	x	-0,236	=	0,056
0,250	x	0,250	=	0,063
-0,084	x	-0,084	=	0,007
-0,099	x	-0,099	=	0,010
-0,158	x	-0,158	=	0,024
	<b>Jumlah</b>		=	<b>0,562</b>

Jumlah (Data2)<sup>2</sup> :

$$\begin{aligned}
 &(-0,040)^2 + (-0,096)^2 + (-0,076)^2 + (0,477)^2 + (0,168)^2 + (-0,050)^2 + (-0,013)^2 + \\
 &(-0,313)^2 + (0,089)^2 + (0,094)^2 + (-0,107)^2 + (-0,236)^2 + (0,250)^2 + (-0,084)^2 + \\
 &(-0,099)^2 + (-0,158)^2 = 0,562
 \end{aligned}$$

Pembagian :

$$\frac{0.562}{0.562} = 1 \quad \text{Sehingga diperoleh nilai koefisien korelasinya adalah 1.}$$

- Komponen *Imajiner*

<u>Data 1</u>		<u>Data 2</u>	=	
-0,244	x	-0,244	=	0,060
-0,582	x	-0,582	=	0,339
-0,021	x	-0,021	=	0,000
-0,020	x	-0,020	=	0,000
-0,144	x	-0,144	=	0,021
-0,112	x	-0,112	=	0,013
-0,267	x	-0,267	=	0,071
0,313	x	0,313	=	0,098
-0,275	x	-0,275	=	0,076
-0,275	x	-0,275	=	0,076
-0,120	x	-0,120	=	0,014
0,084	x	0,084	=	0,007
-0,119	x	-0,119	=	0,014
-0,901	x	-0,901	=	0,812
0,099	x	0,099	=	0,010
-0,285	x	-0,285	=	0,081
<b>Jumlah</b>			=	<b>1,692</b>

Jumlah (Data2)<sup>2</sup> :

$$(-0,244)^2 + (-0,582)^2 + (-0,021)^2 + (-0,020)^2 + (-0,144)^2 + (-0,112)^2 + (-0,267)^2 + (0,313)^2 + (-0,275)^2 + (-0,275)^2 + (-0,120)^2 + (0,084)^2 + (-0,119)^2 + (-0,901)^2 + (0,099)^2 + (-0,285)^2 = 1,692$$

Pembagian :

$$\frac{1.692}{1.692} = 1 \quad \text{Sehingga diperoleh nilai koefisien korelasinya adalah 1.}$$

Berdasarkan pengujian perhitungan antara hasil simulasi dengan perhitungan secara matematis manual ternyata mempunyai hasil yang sama. Sehingga dapat

disimpulkan bahwa kanal proses korelasi dan proses pencarian jendela FFT sesuai dengan perancangan sistem. Hasil dari pencarian jendela FFT dapat dilihat pada Gambar 4.12.

Jendela FFT (Komponen Real)								Jendela FFT (Komponen Imajiner)							
-0.158	-0.099	-0.084	0.250	-0.236	-0.107	0.094	0.089	-0.285	0.099	-0.901	-0.119	0.084	-0.120	-0.275	-0.275
-0.313	-0.013	-0.050	0.168	0.477	-0.076	-0.096	-0.040	0.313	-0.267	-0.112	-0.144	0.020	-0.021	-0.582	-0.244
-0.397	-0.727	0.478	-0.301	-0.064	0.023	0.026	-0.267	-0.094	-0.130	0.209	-0.013	-0.038	0.182	-0.223	0.068
0.281	0.088	-0.019	0.540	-0.468	0.244	0.059	0.140	0.281	0.151	-0.117	0.028	-0.234	-0.330	0.305	0.384
-0.467	-0.296	0.348	0.315	0.317	-0.044	-0.468	-0.301	-0.152	-0.005	-0.493	-0.237	-0.253	-0.256	0.383	0.361
0.000	0.033	0.233	-0.140	-0.558	0.252	0.524	0.399	0.063	-0.143	-0.538	-0.167	0.289	-0.060	0.044	0.079
-0.353	0.089	0.200	-0.147	0.233	-0.100	-0.094	-0.534	0.657	0.395	0.472	0.597	-0.042	0.307	-0.099	0.127
0.406	0.071	-0.105	-0.331	0.299	-0.338	-0.046	0.161	-0.031	0.005	0.231	-0.047	-0.075	-0.305	0.196	0.103

Gambar 4.12 Tampilan hasil pencarian jendela FFT

### 4.3 Analisis Hasil Percobaan

Pada sub bab ini dibahas tentang fenomena yang terjadi pada pencarian jendela FFT yaitu mengenai pengaruh derau kanal AWGN terhadap proses dan hasil dari pencarian jendela FFT untuk sistem OFDM. Setelah itu dianalisis mengenai lamanya waktu pencarian jendela FFT, tingkat keberhasilan, tingkat kecermatan sistem dalam pencarian jendela FFT dan terkecohnya penentuan hasil jendela FFT yang diperoleh.

#### 4.3.1 Pengaruh Daya Derau Terhadap Waktu Pencarian Jendela FFT

Percobaan ini dilakukan untuk meneliti pengaruh besarnya daya derau berbagai nilai terhadap waktu yang dibutuhkan untuk mencari dan menemukan jendela FFT pada sistem OFDM yaitu pada bagian penerima. Waktu disini dalam satuan bingkai (*frame*), sehingga berapa kali proses pencuplikan bingkai (*frame*) untuk mencari dan menentukan data mana yang merupakan jendela FFT, tentunya dengan variasi besarnya daya derau yang digunakan. Adapun parameter masukan percobaan ini adalah :

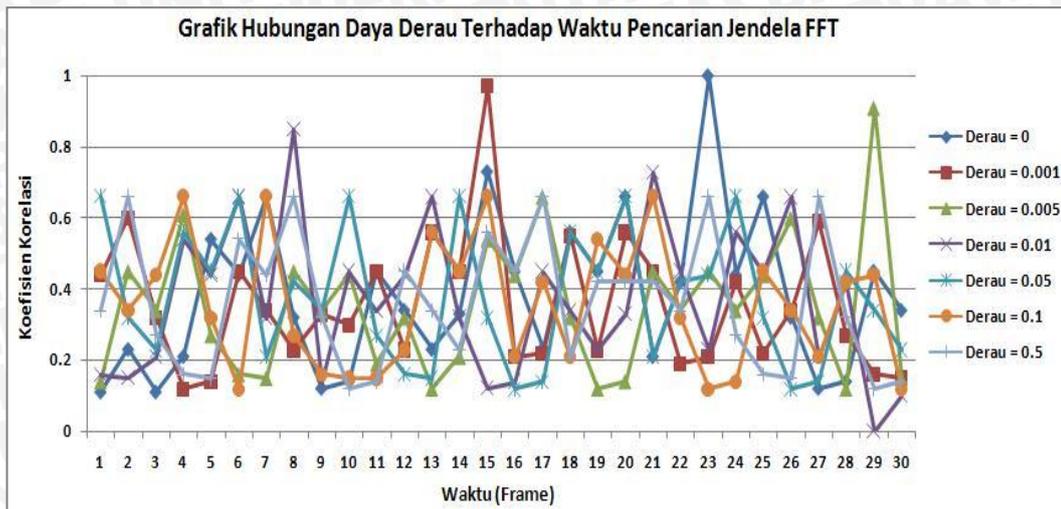
1. Jumlah titik IFFT/FFT (N) adalah 64 titik.
2. Modulasi Digital adalah 16-QAM.
3. Interval *Cyclic Prefix* adalah  $\frac{1}{4}$ .
4. Kanal transmisi dipengaruhi AWGN dengan daya derau bervariasi yaitu 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt.
5. Batas koefisien korelasi adalah 0.8.
6. Interval waktu adalah 1000 ms.

**Tabel 4.6 Hasil percobaan pengaruh daya derau yang bervariasi terhadap nilai koefisien korelasi yang dihasilkan.**

Koefisien Korelasi Bingkai ( <i>Frame</i> ) ke-	Daya Derau						
	0	0.001	0.005	0.01	0.05	0.1	0.5
1	0.11	0.44	0.14	0.16	0.66	0.45	0.34
2	0.23	0.6	0.45	0.15	0.32	0.34	0.66
3	0.11	0.32	0.34	0.21	0.23	0.44	0.27
4	0.21	0.12	0.61	0.54	0.56	0.66	0.16
5	0.54	0.14	0.27	0.44	0.45	0.32	0.15
6	0.44	0.45	0.16	0.66	0.66	0.12	0.54
7	0.66	0.34	0.15	0.32	0.21	0.66	0.44
8	0.32	0.23	0.45	0.85	0.42	0.27	0.66
9	0.12	0.33	0.34	0.14	0.34	0.16	0.32
10	0.14	0.3	0.44	0.45	0.66	0.15	0.12
11	0.45	0.45	0.19	0.34	0.27	0.15	0.14
12	0.34	0.23	0.32	0.44	0.16	0.23	0.45
13	0.23	0.56	0.12	0.66	0.15	0.56	0.34
14	0.33	0.45	0.21	0.32	0.66	0.45	0.23
15	0.73	0.97	0.54	0.12	0.32	0.66	0.56
16	0.45	0.21	0.44	0.14	0.12	0.21	0.45
17	0.23	0.22	0.66	0.45	0.14	0.42	0.66
18	0.56	0.55	0.32	0.34	0.56	0.21	0.21
19	0.45	0.23	0.12	0.23	0.45	0.54	0.42
20	0.66	0.56	0.14	0.33	0.66	0.44	0.42
21	0.21	0.45	0.45	0.73	0.21	0.66	0.42
22	0.42	0.19	0.34	0.45	0.42	0.32	0.34
23	1	0.21	0.45	0.23	0.44	0.12	0.66
24	0.44	0.42	0.34	0.56	0.66	0.14	0.27
25	0.66	0.22	0.44	0.45	0.32	0.45	0.16
26	0.32	0.34	0.6	0.66	0.12	0.34	0.15
27	0.12	0.59	0.32	0.21	0.14	0.21	0.66
28	0.14	0.27	0.12	0.42	0.45	0.42	0.32
29	0.45	0.16	0.91	0	0.34	0.44	0.12
30	0.34	0.15	0.15	0.1	0.23	0.12	0.14

Berdasarkan Tabel hasil percobaan nilai koefisien korelasi yang ditampilkan pada Tabel 4.6. Pengaturan Daya derau AWGN dibuat bervariasi antara 0.001 hingga 0.5 Watt, hal ini dilakukan untuk mengetahui seberapa besar pengaruh daya derau AWGN terhadap waktu pencarian jendela FFT. Dilakukan 30 kali pengamatan agar dapat terlihat pengaruhnya. Untuk mengukur waktu dalam hal ini dengan mengamati berapa bingkai (*frame*) proses pencuplikan dan pengkorelasiannya itu terjadi sampai didapatkan nilai koefisien korelasi  $\geq 0.8$ .

Agar lebih mudah dimengerti maka dibuat grafik hasil percobaan dari Tabel 4.6 tersebut yang dapat dilihat pada Gambar 4.13.



**Gambar 4.13** Pengaruh daya derau terhadap waktu pencarian jendela FFT

Berdasarkan Gambar 4.13 dapat diketahui mengenai pengaruh daya derau AWGN terhadap waktu pencarian jendela FFT yaitu saat ditemukan nilai koefisien korelasi lebih besar dari 0.8. Terlihat bahwa waktu yang dibutuhkan untuk masing-masing derau acak dan tidak linear. Sehingga dapat disimpulkan bahwa semakin besar daya derau yang mempengaruhi kanal AWGN tidak berpengaruh terhadap waktu yang dibutuhkan untuk pencarian jendela FFT. Hal ini disebabkan karena data yang diterima oleh bagian penerima dilakukan proses pencarian dengan cara pencuplikan secara acak, sehingga waktu yang dibutuhkan untuk pencarian jendela FFT tergantung dimana saat mencuplik datanya. Sehingga secara singkatnya dapat disimpulkan bahwa derau tidak berpengaruh terhadap waktu pencarian jendela FFT.

#### 4.3.2 Pengaruh Daya Derau Terhadap Nilai Koefisien Korelasi

Percobaan ini dilakukan untuk meneliti pengaruh besarnya daya derau berbagai nilai terhadap nilai koefisien korelasi yang diperoleh dalam proses korelasi silang pada simulasi pencarian jendela FFT sistem OFDM. Adapun parameter masukan percobaan ini adalah :

1. Jumlah titik IFFT/FFT (N) adalah 64 titik.
2. Modulasi Digital adalah 16-QAM.
3. Interval *Cyclic Prefix* adalah  $\frac{1}{4}$ .

4. Kanal transmisi dipengaruhi AWGN dengan daya derau bervariasi yaitu 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt.
5. Batas Koefisien Korelasi adalah 0.8.
6. Interval waktu adalah 1000 ms.

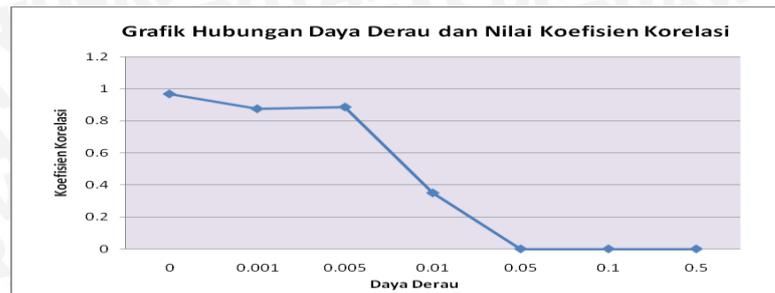
Berdasarkan Tabel hasil percobaan nilai koefisien korelasi yang ditampilkan pada Tabel 4.7. Daya derau AWGN dibuat bervariasi antara 0.001 hingga 0.5 Watt, hal ini dilakukan untuk mengetahui seberapa besar pengaruh daya derau AWGN terhadap nilai koefisien yang dihasilkan setiap kali proses pencarian jendela FFT yang berhasil dengan menggunakan metode korelasi silang. Dilakukan 10 kali percobaan agar dapat terlihat pengaruhnya. Kemudian dicari nilai rata-ratanya.

**Tabel 4.7 Hasil percobaan pengaruh daya derau terhadap nilai koefisien korelasi**

Koef Korelasi Percobaan Berhasil ke-	Daya Derau						
	0	0.001	0.005	0.01	0.05	0.1	0.5
1	1	0.94	0.96	0.87	0	0	0
2	1	0.8	0.98	0.85	0	0	0
3	1	0.8	0.91	0.91	0	0	0
4	0.85	0.93	0.9	0	0	0	0
5	1	0.91	0.91	0	0	0	0
6	1	0.92	0.88	0	0	0	0
7	1	0.91	0.83	0	0	0	0
8	0.95	0.85	0.8	0.87	0	0	0
9	1	0.9	0.84	0	0	0	0
10	0.89	0.8	0.86	0	0	0	0
Rata-rata	0.969	0.876	0.887	0.35	0	0	0

Pada proses perhitungan nilai koefisien korelasi, akan ada dua logika, yang pertama yaitu jika nilai koefisien korelasinya  $\geq 0.8$  yang merupakan batas koefisien korelasi maka ada korelasi yang kuat antar data yang dibandingkan tersebut atau dengan kata lain kedua data tersebut mirip, karena jika dipengaruhi kanal AWGN, data yang diterima pada bagian penerima akan berbeda dengan yang dikirimkan. Yang kedua adalah jika nilai koefisien korelasinya  $< 0.8$  maka tidak ada korelasi yang kuat antar data yang dibandingkan tersebut.

Agar lebih mudah dimengerti maka dibuat grafik hasil percobaan dari Tabel 4.7 tersebut yang dapat dilihat pada Gambar 4.14.



**Gambar 4.14** Pengaruh daya derau terhadap nilai koefisien korelasi

Berdasarkan Gambar 4.14 dapat diketahui mengenai pengaruh daya derau AWGN terhadap hasil perhitungan korelasi silang yaitu mengenai nilai koefisien korelasi yang dihasilkan, sehingga dapat disimpulkan bahwa semakin besar daya derau yang mempengaruhi kanal AWGN maka semakin kecil nilai koefisien korelasi sehingga dengan kata lain semakin besar daya AWGN maka semakin kecil kemungkinan berhasil dalam pencarian jendela FFT, hal ini karena nilai koefisien korelasinya semakin ke arah 0 (nol). Berdasarkan Gambar 5.12 terlihat bahwa mulai dari penggunaan daya derau 0.05 Watt relatif dengan SNR 19.5 dB sudah tidak dapat menemukan jendela FFT karena nilai koefisien korelasinya 0 (nol). Batas kestabilan sistem pencarian jendela FFT yaitu pada penggunaan daya derau maksimum 0.005 Watt relatif dengan SNR 29.5 dB.

Pada simulasi sistem OFDM ini, nilai maksimum simbol OFDM yang ditransmisikan adalah 3. Karena nilai tersebut dapat dianggap sebagai tegangan, maka dayanya adalah  $3^2 = 9$  dan daya rata-ratanya dapat diasumsikan setengah dari dayanya yaitu 4,5, sehingga  $P_{\text{signal}}$  adalah 4,5 Watt. Variasi  $P_{\text{derau}}$  yang digunakan pada simulasi sistem OFDM ini adalah 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt sehingga berdasarkan hal tersebut, perhitungan relatif SNR (dB) yang digunakan untuk simulasi ini yaitu

$$SNR (dB) = 10^{10} \log \frac{P_{\text{signal}}}{P_{\text{derau}}} \quad (4.1)$$

- Untuk  $P_{\text{derau}} = 0.001$   
 $SNR(dB) = 10^{10} \log 4.5 / 0.001 = 36.5 \text{ dB}$
- Untuk  $P_{\text{derau}} = 0.005$   
 $SNR(dB) = 10^{10} \log 4.5 / 0.005 = 29.5 \text{ dB}$
- Untuk  $P_{\text{derau}} = 0.01$   
 $SNR(dB) = 10^{10} \log 4.5 / 0.01 = 26.5 \text{ dB}$
- Untuk  $P_{\text{derau}} = 0.05$

$$SNR(dB) = 10^{10} \log 4.5 / 0.05 = 19.5 \text{ dB}$$

- Untuk  $P_{\text{derau}} = 0.1$

$$SNR(dB) = 10^{10} \log 4.5 / 0.1 = 16.5 \text{ dB}$$

- Untuk  $P_{\text{derau}} = 0.5$

$$SNR(dB) = 10^{10} \log 4.5 / 0.5 = 9.5 \text{ dB}$$

### 4.3.3 Pengaruh Daya Derau Terhadap Keberhasilan Pencarian Jendela FFT

Percobaan ini dilakukan untuk meneliti pengaruh daya derau kanal AWGN terhadap keberhasilan dalam pencarian jendela FFT. Untuk menganalisisnya memerlukan minimal 10 kali percobaan untuk dapat melihat besarnya pengaruh daya derau terhadap keberhasilan pencarian jendela FFT tersebut. Untuk melihat tingkat keberhasilan pencarian jendela FFT dapat diamati melalui kestabilan nilai koefisien korelasinya.

Adapun parameter masukan percobaan ini adalah :

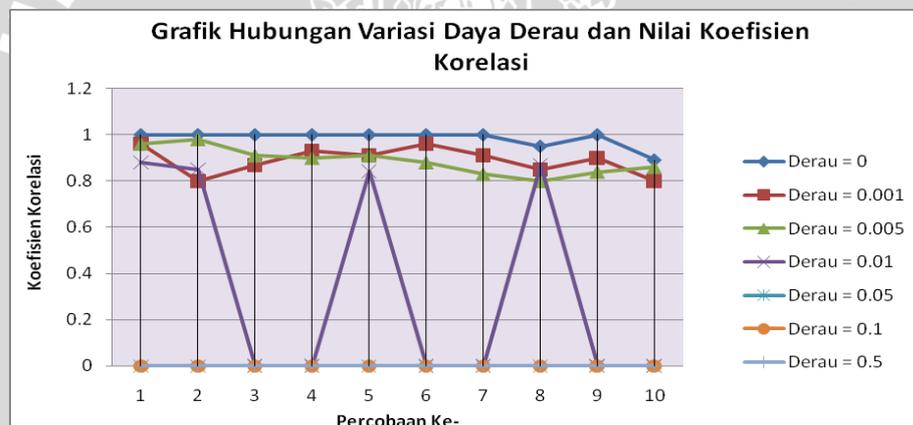
1. Jumlah titik IFFT/FFT (N) adalah 64 titik.
2. Modulasi Digital adalah 16-QAM.
3. Interval *Cyclic Prefix* adalah  $\frac{1}{4}$ .
4. Kanal transmisi dipengaruhi AWGN dengan daya derau bervariasi yaitu 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt.
5. Batas koefisien korelasi adalah 0.8.
6. Interval waktu adalah 1000 ms..

Percobaan ini hampir sama dengan percobaan sebelumnya, namun untuk nilai koefisien korelasi pada variasi besarnya derau tidak perlu dicari rata-ratanya. Tabel hasil percobaan nilai hasil koefisien korelasi dapat dilihat pada Tabel 4.8. Berdasarkan Tabel tersebut terlihat bahwa mulai dari besar daya derau 0.05 Watt sampai 0.5 Watt nilai koefisien korelasinya 0 semua dari 10 kali percobaan, hal ini berarti mulai dari besar daya derau 0.05 Watt sampai 0.5 Watt sama sekali tidak ada peluang keberhasilan dalam penentuan hasil pencarian jendela FFT, dalam arti lain jendela FFT tidak akan pernah bisa ditemukan karena data yang diterima sangat rusak akibat derau. Sedangkan pada daya derau 0.01 Watt, keberhasilan pencarian jendela FFT tidak stabil, terkadang bisa ditemukan namun terkadang tidak bisa ditemukan. Agar lebih mudah dimengerti maka dibuat grafik hasil percobaan dari Tabel tersebut yang dapat dilihat pada Gambar 4.15. Berdasarkan grafik tersebut dapat disimpulkan bahwa maksimal besarnya daya derau

pada simulasi ini yaitu sebesar 0.005 Watt untuk dapat stabil dalam keberhasilan pencarian jendela FFT.

**Tabel 4.8 Hasil percobaan tingkat keberhasilan pencarian jendela FFT**

Koefisien Korelasi Percobaan ke-	Derau AWGN						
	0	0.001	0.005	0.01	0.05	0.1	0.5
1	1	0.96	0.96	0.88	0	0	0
2	1	0.8	0.98	0.85	0	0	0
3	1	0.87	0.91	0	0	0	0
4	1	0.93	0.9	0	0	0	0
5	1	0.91	0.91	0.84	0	0	0
6	1	0.96	0.88	0	0	0	0
7	1	0.91	0.83	0	0	0	0
8	0.95	0.85	0.8	0.87	0	0	0
9	1	0.9	0.84	0	0	0	0
10	0.89	0.8	0.86	0	0	0	0



**Gambar 4.15 Pengaruh daya derau terhadap hasil koefisien korelasi**

#### 4.3.4 Pengaruh Daya Derau Terhadap Kecermatan Pencarian Jendela FFT

Percobaan ini dilakukan untuk meneliti pengaruh besarnya daya derau kanal AWGN terhadap ketepatan hasil pencarian jendela FFT. Apakah data hasil pencarian tersebut benar merupakan jendela FFT atau tidak, karena mungkin bisa saja itu bukan merupakan jendela FFT, dalam arti lain sistem terkecoh. Adapun parameter masukan percobaan ini adalah :

1. Jumlah titik IFFT/FFT (N) adalah 64 titik.
2. Modulasi Digital adalah 16-QAM.
3. Interval *Cyclic Prefix* adalah  $\frac{1}{4}$ .
4. Kanal transmisi tidak dipengaruhi AWGN (Daya Derau = 0)

5. Batas Koefisien Korelasi adalah 0.8.
6. Interval waktu adalah 1000 ms.

Pada percobaan ini tidak digunakan daya derau dari kanal AWGN. Hal ini untuk melihat apakah hasil korelasi yang dihasilkan selalu 1 jika tidak dipengaruhi oleh derau, karena jika tanpa derau hasil korelasinya harus 1, hal ini disebabkan data yang dikirim pasti akan sama dengan data yang diterima. Jika hasil korelasinya kurang dari 1, maka ada ketidaktepatan dalam hasil jendela FFT yang didapat yang ternyata sebenarnya hasil jendela FFT yang didapat tersebut bukan merupakan jendela FFT. Untuk percobaannya dilakukan proses pencuplikan dan pengkorelasiian secara terus menerus untuk beberapa kali percobaan. Hal ini karena analisisnya ditujukan untuk melihat apakah ada kemungkinan adanya kesalahan sistem.

**Tabel 4.9 Hasil percobaan tingkat kecermatan pencarian jendela FFT**

Koefisien Korelasi	Derau AWGN
Pencuplikan Berhasil Ke-	0
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	0.95
9	1
10	0.89

Tabel hasil percobaan berupa hasil koefisien korelasi dapat dilihat pada Tabel 4.9. Agar lebih mudah dimengerti maka dibuat grafik hasil percobaan dari Tabel tersebut yang dapat dilihat pada Gambar 4.16. Berdasarkan grafik tersebut dapat disimpulkan bahwa pada proses pencarian jendela FFT tanpa adanya pengaruh derau ( Daya derau = 0) ternyata hasil koefisien korelasi dalam proses pencarian jendela FFT tidak mutlak selalu 1. Karena sistem digunakan untuk kanal transmisi yang dipengaruhi kanal AWGN, maka data yang dikirim akan berbeda dengan yang diterima sehingga nilai koefisien korelasinya  $< 1$ . Maka dilakukan pengaturan batas koefisien korelasi yaitu  $\geq 0.8$ , sehingga sistem akan membaca bahwa ada korelasi yang kuat antara data yang dibandingkan jika

hasil korelasi silangnya  $\geq 0.8$ . Sedangkan jika Daya derau = 0, maka nilai koefisien korelasinya harus = 1.



**Gambar 4.16** Ketepatan nilai koefisien korelasi

Percobaan tersebut dapat disimpulkan juga bahwa semakin besar daya derau yang mempengaruhi sinyal maka akan semakin besar pula terjadinya kemungkinan kesalahan, dalam arti lain semakin besar pula terjadinya ketidaktepatan penentuan hasil jendela FFT. Hal tersebut menentukan tingkat kecermatan sistem pencarian jendela FFT.

#### 4.3.5 Pengaruh Batas Koefisien Korelasi Terhadap Waktu Pencarian Jendela FFT

Percobaan ini dilakukan untuk meneliti pengaruh penentuan batas minimal nilai koefisien korelasi terhadap lamanya waktu pencarian jendela FFT dengan variasi daya derau. Adapun parameter masukan percobaan ini adalah :

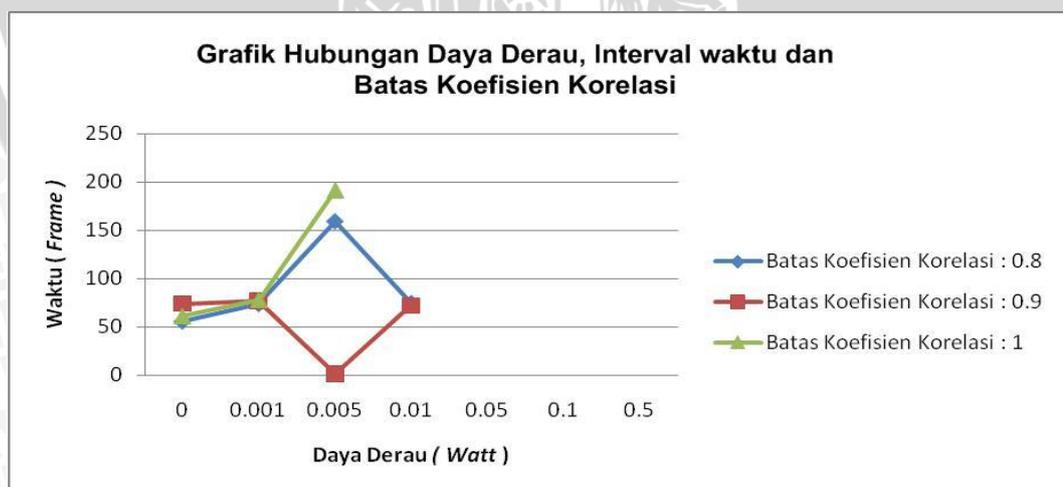
1. Jumlah titik IFFT/FFT (N) adalah 64 titik.
2. Modulasi Digital adalah 16-QAM.
3. Interval *Cyclic Prefix* adalah  $\frac{1}{4}$ .
4. Kanal transmisi dipengaruhi AWGN dengan daya derau bervariasi yaitu 0.001, 0.005, 0.01, 0.05, 0.1, 0.5 Watt.
5. Batas Koefisien Korelasi adalah 0.8, 0.9, 1.
6. Interval waktu adalah 1000 ms.

Pada percobaan ini digunakan daya derau dari kanal AWGN dan variasi batas koefisien korelasi. Hal ini untuk melihat pengaruh pencarian jendela FFT. Tabel hasil percobaan berupa lamanya waktu pencarian jendela FFT dapat dilihat pada Tabel 4.10. Grafik hasil percobaan agar bisa dimengerti maka hasil percobaan dari Tabel tersebut yang dapat dilihat pada Gambar 4.17. Berdasarkan grafik tersebut dapat disimpulkan bahwa pada proses pencarian jendela FFT dengan variasi pengaruh besarnya daya derau ternyata waktu yang dibutuhkan untuk pencarian jendela FFT tidak linear dengan

pengaturan batas koefisien korelasi 0.8 sampai 1. Sehingga dapat disimpulkan bahwa besarnya pengaturan batas nilai koefisien korelasi tidak mempengaruhi waktu pencarian jendela FFT, namun mempengaruhi keberhasilan pencarian jendela FFT, hal ini terlihat pada saat daya derau 0.01 Watt, dengan pengaturan batas minimum koefisien korelasi 1 maka jendela FFT tidak berhasil ditemukan. Selain itu juga mempengaruhi kecermatan hasil pencarian jendela FFT, hal ini terlihat pada saat daya derau 0.001 Watt dan 0.005 Watt, dengan pengaturan batas minimum koefisien korelasi 1 ternyata masih dapat ditemukan, sedangkan jika sistem dipengaruhi AWGN maka tidak mungkin hasil korelasinya 1 karena data yang dikirim tidak akan sama persis dengan data yang diterima. Maka dapat dikatakan bahwa sistem terkecoh.

**Tabel 4.10 Hasil percobaan pengaruh batas koefisien korelasi terhadap waktu pencarian jendela FFT**

Waktu (Frame) Saat Daya Derau	Batas Koefisien Korelasi		
	0.8	0.9	1
0	56	74	61
0.001	74	77	78
0.005	159	1	191
0.01	75	72	-
0.05	-	-	-
0.1	-	-	-
0.5	-	-	-



**Gambar 4.17 Pengaruh batas koefisien korelasi terhadap waktu pencarian jendela FFT**

Maka dari percobaan tersebut, secara garis besar dapat disimpulkan bahwa penentuan batas nilai koefisien korelasi berpengaruh terhadap keberhasilan dan kecermatan pencarian jendela FFT, namun tidak berpengaruh terhadap waktu.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian mengenai pencarian jendela FFT untuk sistem OFDM, didapatkan beberapa kesimpulan sebagai berikut :

1. Daya derau dan batas koefisien korelasi berpengaruh terhadap tingkat keberhasilan dan kecermatan, namun tidak berpengaruh terhadap waktu pada sistem pencarian jendela FFT.
2. Nilai koefisien korelasi = 1 untuk daya derau = 0.
3. Dibutuhkan daya 0.005 Watt untuk dapat stabil dalam keberhasilan pencarian jendela FFT.
4. Batas Kuantitatif SNR(dB) = 19.5 dB menghasilkan nilai koefisien korelasi 0 ( nol ) sudah tidak dapat menemukan jendela FFT.
5. Kestabilan sistem pencarian jendela FFT pada batas minimum kuantitatif SNR(dB) = 29.5 dB.

#### 5.2 Saran

Saran yang dapat digunakan untuk penelitian selanjutnya adalah perlu diteliti lebih lanjut mengenai pencarian jendela FFT untuk sistem OFDM dengan menggunakan :

1. Perlu dilakukan penelitian selanjutnya untuk sistem pencarian jendela FFT tidak hanya menggunakan metode korelasi saja, hal ini untuk memperbesar tingkat kecermatan dan kehandalan sistem dalam pencarian jendela FFT.
2. Dilakukan penelitian selanjutnya untuk jumlah titik IFFT/FFT yang lebih kompleks seperti 128 titik, 256 titik, 512 titik, 1024 titik.
3. Berbagai macam modulasi digital QAM seperti 32-QAM, 64-QAM, 128-QAM dan 256-QAM untuk melihat pengaruh waktu pencarian jendela FFT.

## DAFTAR PUSTAKA

- Di, Zhang. *Performance Analysis and Comparison of OFDM Based Packet Transmission System with QAM Modulation*. Communication System Group, Dept. of signals and Systems, Chalmers University of Technology. 2004
- Faisal, Muhammad. *Pengaruh Panjang Cyclic Prefix Terhadap Kinerja Sistem OFDM Pada WiMAX*, Tugas Akhir Sarjana Fakultas Teknik Universitas Sumatera Utara, Medan, 2009.
- Hanzo, L., and T. Keller, 2006, *OFDM and MC-CDMA*, John Wiley & Sons Ltd., Chichester.
- [http://en.wikipedia.org/wiki/Quadrature\\_amplitude\\_modulation](http://en.wikipedia.org/wiki/Quadrature_amplitude_modulation), Maret 2011.
- <http://en.wikipedia.org/wiki/OFDM>, Maret 2011.
- Ludmen, Lonnie. 1987, *Fundamentals of Digital Signal Processing*, John Wiley & Sons., New York.
- Tanudjaja, Harlianto. *Pengolahan Sinyal Digital & Sistem Pemrosesan Sinyal*, 2007.
- Taub, H., and Schilling, L. Donald., 1986, *Principles Of Communication Systems*, 2<sup>nd</sup> edition, McGraw-Hill Inc., New York.
- Tocci, R. J., and Neal S. Widmer, 1998, *Digital Systems Principles and Applications*, 7<sup>th</sup> edition, Prentice-Hall Inc., New Jersey.
- Walpole, E. Ronald., and Myers H. Raymond, 1995, *Probability and Statistics for Engineers and Scientists*, 4<sup>th</sup> edition, Macmillan Publishing Co., Inc., Virginia.
- Winch, G. Robert., 1993, *Telecommunication Transmission Systems*, McGraw-Hill Inc., New York.

## LAMPIRAN

### Listing Program

```
unit windowing_fft;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, Grids, StdCtrls, math, TeEngine, Series, ExtCtrls, TeeProcs,
  Chart, ComCtrls, TabNotBk, jpeg;

type
  TForm1 = class(TForm)
    start: TButton;
    Stop: TButton;
    Panel1: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel10: TPanel;
    Panel17: TPanel;
    PageControl1: TPageControl;
    PageControl3: TPageControl;
    PageControl4: TPageControl;
    PageControl5: TPageControl;
    PageControl6: TPageControl;
    PageControl7: TPageControl;
    TabSheet2: TTabSheet;
    TabSheet10: TTabSheet;
    TabSheet3: TTabSheet;
    TabSheet12: TTabSheet;
    TabSheet4: TTabSheet;
    TabSheet13: TTabSheet;
    TabSheet6: TTabSheet;
    TabSheet8: TTabSheet;
    TabSheet5: TTabSheet;
    TabSheet14: TTabSheet;
    TabSheet17: TTabSheet;
    Memo1: TMemo;
    Memo2: TMemo;
    Memo3: TMemo;
    Memo4: TMemo;
    Real: TLabel;
    imajiner: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label11: TLabel;
    Memo5: TMemo;
    Label7: TLabel;
    Memo6: TMemo;
```

```
Label12: TLabel;  
Button1: TButton;  
Memo7: TMemo;  
Memo8: TMemo;  
Ambil: TButton;  
Label15: TLabel;  
Label16: TLabel;  
StringGrid3: TStringGrid;  
StringGrid4: TStringGrid;  
StringGrid5: TStringGrid;  
StringGrid6: TStringGrid;  
Chart1: TChart;  
BarSeries3: TBarSeries;  
Chart2: TChart;  
BarSeries4: TBarSeries;  
Chart4: TChart;  
Series2: TBarSeries;  
Chart5: TChart;  
Series1: TBarSeries;  
Chart6: TChart;  
Series4: TBarSeries;  
Chart7: TChart;  
Series3: TBarSeries;  
Chart8: TChart;  
Series6: TBarSeries;  
Chart3: TChart;  
BarSeries5: TBarSeries;  
StringGrid1: TStringGrid;  
StringGrid2: TStringGrid;  
Label19: TLabel;  
Label20: TLabel;  
Panel5: TPanel;  
ComboBox1: TComboBox;  
ComboBox2: TComboBox;  
Label13: TLabel;  
Label14: TLabel;  
Edit1: TEdit;  
Panel7: TPanel;  
Panel9: TPanel;  
Label3: TLabel;  
Memo9: TMemo;  
Memo10: TMemo;  
TabSheet1: TTabSheet;  
StringGrid8: TStringGrid;  
Edit2: TEdit;  
StringGrid9: TStringGrid;  
Edit4: TEdit;  
Edit3: TEdit;  
Edit5: TEdit;  
Label23: TLabel;  
Label24: TLabel;  
Memo11: TMemo;  
Memo12: TMemo;  
Panel6: TPanel;  
Panel11: TPanel;  
Label4: TLabel;  
Edit6: TEdit;  
Edit7: TEdit;  
Label10: TLabel;  
Edit8: TEdit;  
Label17: TLabel;
```



```

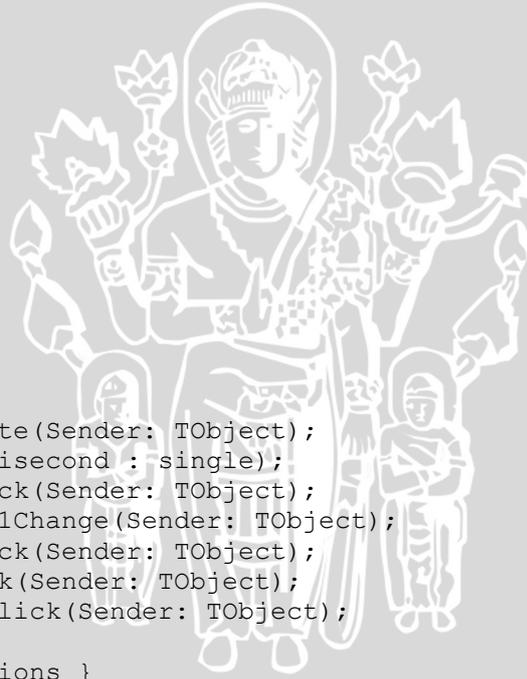
Edit9: TEdit;
Label18: TLabel;
Edit10: TEdit;
Edit11: TEdit;
Panel14: TPanel;
Label22: TLabel;
Label25: TLabel;
Panel15: TPanel;
ComboBox3: TComboBox;
Label21: TLabel;
TabSheet7: TTabSheet;
Memo13: TMemo;
Memo14: TMemo;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label32: TLabel;
Label31: TLabel;
Label33: TLabel;
Image1: TImage;
Panel12: TPanel;
Panel8: TPanel;
Panel13: TPanel;
Panel16: TPanel;
Panel2: TPanel;
Panel18: TPanel;
Image8: TImage;
Image2: TImage;
Image3: TImage;
Image4: TImage;
Image5: TImage;
Image6: TImage;
Image7: TImage;

procedure FormCreate(Sender: TObject);
procedure Wait(milisecond : single);
procedure startClick(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure AmbilClick(Sender: TObject);
procedure StopClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
procedure Awal;
procedure SGrid;
procedure jendela1;
procedure jendela2;
public
{ Public declarations }
end;

var
Form1      : TForm1;
input_ifft,
out_ifft,cp,
cp_awgn,awgn      : array[0..3,0..10000] of extended;
tunda,data_simpan,temp6      : Integer;

Berhenti,jendela      : boolean;

```



```

temp0,s          : String;
input           : array[0..3, 0..10000] of extended;
simpan,data,data1,hitung0,kuadrat,temp10 : array[0..3, 0..10000] of
extended;

```

implementation

```
{$R *.dfm}
```

```
function indexing_out(index_fft:integer;titik_fft:integer):integer;
//fungsi untuk peng-indeks-an pada output ifft dan fft
var
```

```
    k_ndx,dec_ndx,temp_ndx,bit_ndx:integer;
    bin_ndx:array [0..20] of integer;
```

```
begin
```

```
    for k_ndx:=0 to 20 do bin_ndx[k_ndx]:=0;
```

```
    k_ndx:=0;
```

```
    dec_ndx:=index_fft;
```

```
    while dec_ndx > 0 do
```

```
        begin
```

```
            bin_ndx[k_ndx]:=dec_ndx mod 2;
```

```
            dec_ndx:=dec_ndx div 2;
```

```
            k_ndx:=k_ndx+1;
```

```
        end;
```

```
    temp_ndx:=0;
```

```
    bit_ndx:=strtoint(floattostr((log2(titik_fft))-1));
```

```
    for k_ndx:=0 to bit_ndx do
```

```
temp_ndx:=temp_ndx+bin_ndx[k_ndx]*strtoint(floattostr(exp((bit_ndx-
k_ndx)*ln(2)))));
```

```
    indexing_out:=temp_ndx;
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
    SGrid;
```

```
end;
```

```
procedure TForm1.SGrid;
```

```
var
```

```
    i : integer;
```

```
begin
```

```
    for i := 0 to 63 do
```

```
        begin
```

```
            with stringgrid1 do
```

```
                begin
```

```
                    cells[0,0]:= ' ' + 'Sub-Kanal';
```

```
                    cells[1,0]:= ' ' + 'Real';
```

```
                    cells[0,i+1]:= ' ' + inttostr(i);
```

```
                end;
```

```
            with stringgrid2 do
```

```
                begin
```

```
                    cells[0,0]:= ' ' + 'Sub-Kanal';
```

```
                    cells[1,0]:= ' ' + 'Imajiner';
```

```
                    cells[0,i+1]:= ' ' + inttostr(i);
```

```
                end;
```

```
            with stringgrid3 do
```

```
                begin
```

```

cells[0,0]:= ' ' + 'Sub-Kanal';
cells[1,0]:= ' ' + 'Real';
cells[0,i+1]:= ' ' + inttostr(i);
end;

with stringgrid4 do
begin
cells[0,0]:= ' ' + 'Sub-Kanal';
cells[1,0]:= ' ' + 'Imajiner';
cells[0,i+1]:= ' ' + inttostr(i);
end;
end;

for i := 0 to 79 do
begin
with stringgrid5 do
begin
cells[0,0]:= ' ' + 'Sub-Kanal';
cells[1,0]:= ' ' + 'Real';
cells[0,i+1]:= ' ' + inttostr(i);
end;

with stringgrid6 do
begin
cells[0,0]:= ' ' + 'Sub-Kanal';
cells[1,0]:= ' ' + 'Imajiner';
cells[0,i+1]:= ' ' + inttostr(i);
end;
end;

for i := 0 to 15 do
begin
with stringgrid8 do
begin
cells[0,0]:= ' ' + 'CP';
cells[0,i+1]:= ' ' + inttostr(i);
cells[1,0]:= ' ' + 'A';
cells[2,0]:= ' ' + 'B';
cells[3,0]:= ' ' + 'A x B';
cells[4,0]:= ' ' + 'B x B';

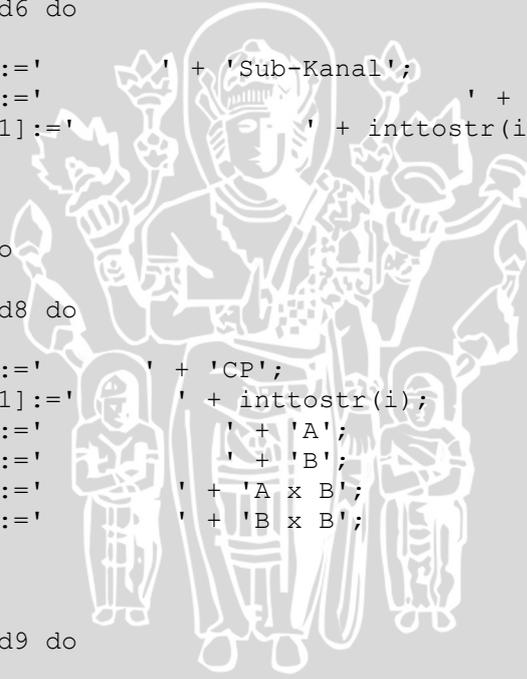
end;

with stringgrid9 do
begin
cells[0,0]:= ' ' + 'CP';
cells[0,i+1]:= ' ' + inttostr(i);
cells[1,0]:= ' ' + 'A';
cells[2,0]:= ' ' + 'B';
cells[3,0]:= ' ' + 'A x B';
cells[4,0]:= ' ' + 'B x B';

end;
end;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
If ComboBox1.Text='Ya' then
begin
ComboBox2.Enabled:=True;

```



```

        ComboBox2.Color:=clWindow;
    end else begin
        ComboBox2.Enabled:=False;
        ComboBox2.Text:='';
        ComboBox2.Color:=$00E9FDFE;
    end;
end;

procedure TForm1.Awal;
var
    m, subkanal,i,j,n_fft,n_stage,stage,j2,j3,jw,k,int: integer;
    temp4,tempfft0,tempfft1,std_deviasi           : extended;
    w,infft,outfft                               :array[0..19,0..100] of extended;
begin
    //===== masukan ifft =====//
    randomize;
    for i:=0 to 63 do
    Begin
        input_ifft[0,i]:=random(4);
        if input_ifft[0,i]=0 then input_ifft[0,i]:=-3
        else if input_ifft[0,i]=1 then input_ifft[0,i]:=-1
        else if input_ifft[0,i]=2 then input_ifft[0,i]:=1;

        input_ifft[1,i]:=random(4);
        if input_ifft[1,i]=0 then input_ifft[1,i]:=-3
        else if input_ifft[1,i]=1 then input_ifft[1,i]:=-1
        else if input_ifft[1,i]=2 then input_ifft[1,i]:=1;
    end;

    //===== IFFT =====//
    for i:=0 to 63 do
    begin
        w[0,i]:=cos(i*pi/64);
        w[1,i]:=-1*sin(i*pi/64);
    end;

    //==== Tukar Real & Imaj in_ifft (A + j B >>> B + j A) =====//
    for i:=0 to 63 do
    begin
        temp4:=input_ifft[0,i];
        input_ifft[0,i]:=input_ifft[1,i];
        input_ifft[1,i]:=temp4;
    end;

    //===== infft:=input_ifft =====//
    for i:=0 to 63 do
    begin
        for j:=0 to 1 do
        begin
            infft[j,i]:=input_ifft[j,i];
        end;
    end;

    //===== fft =====//
    n_stage:=strtoint(floattostr(log2(64)));
    n_fft:=64;
    for stage:=1 to n_stage do
    begin
        for i:=1 to strtoint(floattostr(power(2,stage-1))) do
        begin

```

```

do
    for j:=0 to strtoint(floattostr((n_fft/(power(2,stage))-1))
    begin
        j2:=j+strtoint(floattostr((n_fft/(power(2,stage-1))*(i-
        1)))));
        j3:=j2+strtoint(floattostr(n_fft/power(2,stage)));
        jw:=strtoint(floattostr(j*power(2,stage)));

        //A+B
        outfft[0,j2]:=infft[0,j2]+infft[0,j3];
        outfft[1,j2]:=infft[1,j2]+infft[1,j3];

        //A-B
        tempfft0:=infft[0,j2]-infft[0,j3];
        tempfft1:=infft[1,j2]-infft[1,j3];

        //(A-B)*w
        outfft[0,j3]:=tempfft0*w[0,jw]-tempfft1*w[1,jw];
        outfft[1,j3]:=tempfft0*w[1,jw]+tempfft1*w[0,jw];

        //masukan data outut jadi input untuk stage berikutnya
        infft[0,j3]:=outfft[0,j3];
        infft[1,j3]:=outfft[1,j3];
        infft[0,j2]:=outfft[0,j2];
        infft[1,j2]:=outfft[1,j2];
    end;
end;

//===== Indexing Keluaran FFT =====//
for i:=0 to n_fft-1 do
begin
    k:=indexing_out(i,n_fft);
    for j:=0 to 1 do
    begin
        outfft[j,k]:=infft[j,i];
    end;
end;

//===== Tukar Real & imaj lalu dikalikan 1/n_fft =====//
for i:=0 to n_fft-1 do
begin
    out_ifft[0,i]:=outfft[1,i]/n_fft;
    out_ifft[1,i]:=outfft[0,i]/n_fft;
end;

//===== Cyclic Prefix =====//
for i:=0 to 15 do
begin
    cp[0,i]:=out_ifft[0,i+48];
    cp[1,i]:=out_ifft[1,i+48];
end;

for i:=16 to 79 do
begin
    cp[0,i]:=out_ifft[0,i-16];
    cp[1,i]:=out_ifft[1,i-16];
end;

//===== AWGN =====//

```

```

    if ComboBox2.Text = '' then std_deviasi := 0
    else std_deviasi:=sqrt(strtoint(ComboBox2.Text));

    randomize;
    for i:=0 to 79 do
    begin
        awgn[0,i]:=randg(0,std_deviasi);
        awgn[1,i]:=randg(0,std_deviasi);
    end;

    //===== Sinyal + AWGN =====//
    for i:=0 to 79 do
    begin
        for j:=0 to 1 do
        begin
            cp_awgn[j,i]:=cp[j,i]+awgn[j,i];
        end;
    end;
end;

procedure TForm1.Wait(milisecond : single);
var
    pos, h : single;
begin
    pos:= 0.00001;
    h:=milisecond;
    while pos<h do
    begin
        application.ProcessMessages();
        if pos<h then
        begin
            pos:=pos+0.0001;
        end;
    end;
end;

procedure TForm1.startClick(Sender: TObject);
var
    i,temp5,ulang_i,x : integer;
begin
    Berhenti:=False;
    jendela:=False;
    start.Caption:='Ulang';
    start.Enabled:=false;
    button1.Enabled:=false;
    stop.Enabled:=true;
    tunda:=strtoint(Edit1.Text);
    temp5:=0;
    label3.Caption:='0';
    ComboBox1.Enabled:=false;
    ComboBox1.Color:=$00E9FDFE;
    Edit1.Enabled:=false;
    Edit1.Color:=$00E9FDFE;

    memo1.Clear; memo2.Clear; memo3.Clear; memo4.Clear;
    memo5.Clear; memo6.Clear; memo7.Clear; memo8.Clear;
    memo9.Clear; memo10.Clear; memo11.Clear; memo12.Clear;
    edit2.Clear; edit3.Clear; edit4.Clear; edit5.Clear;
    edit6.Clear; edit7.Clear; edit8.Clear; edit9.Clear;
    edit10.Clear; edit11.Clear;
    edit8.Color:=clWindow; edit11.Color:=clWindow;

```

```

for ulang_i:=0 to 80 do
begin
  stringgrid1.Rows[ulang_i].Clear;
  stringgrid2.Rows[ulang_i].Clear;
  stringgrid3.Rows[ulang_i].Clear;
  stringgrid4.Rows[ulang_i].Clear;
  stringgrid5.Rows[ulang_i].Clear;
  stringgrid6.Rows[ulang_i].Clear;
  stringgrid8.Rows[ulang_i].Clear;
  stringgrid9.Rows[ulang_i].Clear;
end;

SGrid;
temp5:=temp5;
Chart1.Series[0].Clear; Chart2.Series[0].Clear;
Chart3.Series[0].Clear; Chart4.Series[0].Clear;
Chart5.Series[0].Clear; Chart6.Series[0].Clear;
Chart7.Series[0].Clear; Chart8.Series[0].Clear;
Chart1.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart2.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart3.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart4.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart5.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart6.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart7.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
Chart8.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
awal;
x:=0;
repeat
  if x = 80 then
  begin
    temp5:=temp5;
    Chart1.Series[0].Clear; Chart2.Series[0].Clear;
    Chart3.Series[0].Clear; Chart4.Series[0].Clear;
    Chart5.Series[0].Clear; Chart6.Series[0].Clear;
    Chart7.Series[0].Clear; Chart8.Series[0].Clear;
    Chart1.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart2.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart3.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart4.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart5.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart6.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart7.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    Chart8.Series[0].AddXY(79,cp[0,0],'',clTeeColor);
    awal;
    x:=0;
  end;
  i:=x;
  temp5:=temp5+1;

  //===== Menampilkan Masukan IFFT =====//
  with stringgrid1 do
  begin
    s:=floattostr(input_ifft[0,i]);
    if length(s) < 2 then s := ' ' + s;
    cells[1,i+1]:= ' ' + s ;
  end;

  with stringgrid2 do

```

```

begin
    s:=floattostr(input_ifft[1,i]);
    if length(s) < 2 then s := ' ' + s;
    cells[1,i+1]:= ' ' + s ;
end;

//===== Menampilkan Keluaran IFFT =====//
with stringgrid3 do
begin
    str(out_ifft[0,i]:0:3,temp0);
    s:=temp0;
    if length(s) < 6 then s := ' ' + s;
    cells[1,i+1]:= ' ' + s ;
end;

with stringgrid4 do
begin
    str(out_ifft[1,i]:0:3,temp0);
    s:=temp0;
    if length(s) < 6 then s := ' ' + s;
    cells[1,i+1]:= ' ' + s ;
end;

//===== Menampilkan Hasil Penambahan CP =====//
with stringgrid5 do
begin
    str(cp[0,i]:0:3,temp0);
    s:=temp0;
    if length(s) < 6 then s := ' ' + s;
    cells[1,i+1]:=' ' + s + ' ' + cells[1,i+1];
    Memo1.Text:=s + ' ' + Memo1.Text;
end;

with stringgrid6 do
begin
    str(cp[1,i]:0:3,temp0);
    s:=temp0;
    if length(s) < 6 then s := ' ' + s;
    cells[1,i+1]:=' ' + s + ' ' + cells[1,i+1];
    Memo2.Text:=s + ' ' + Memo2.Text;
end;

//=== Menampilkan Hasil Penambahan Sinyal + AWGN ===//
str(cp_awgn[0,i]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Memo3.Text:=s + ' ' + Memo3.Text;

str(cp_awgn[1,i]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Memo4.Text:=s + ' ' + Memo4.Text;

//===== Menampilkan Chart CP =====//
Chart1.Series[0].AddXY(i,cp[0,i],',',clTeeColor);
Chart2.Series[0].AddXY(i,cp[1,i],',',clTeeColor);
Chart3.Series[0].AddXY(i,cp[0,i],',',clTeeColor);
Chart4.Series[0].AddXY(i,cp[1,i],',',clTeeColor);

//===== Menampilkan Chart AWGN =====//
Chart5.Series[0].AddXY(i,awgn[0,i],',',clTeeColor);

```

```

Chart6.Series[0].AddXY(i,awgn[1,i],'',clTeeColor);

//===== Menampilkan Chart CP + AWGN =====//
Chart7.Series[0].AddXY(i,cp_awgn[0,i],'',clTeeColor);
Chart8.Series[0].AddXY(i,cp_awgn[1,i],'',clTeeColor);

//===== Record data =====//
data_simpan:=temp5;
if data_simpan = 80 then
begin
    ambil.Enabled:=true;
end;

data[0,temp5]:= cp_awgn[0,i];
data[1,temp5]:= cp_awgn[1,i];

if jendela = true then
begin
    jendela2;
end;

x:=x+1;
wait (tunda);
Application.ProcessMessages();
until Berhenti=True;
end;

procedure TForm1.AmbilClick(Sender: TObject);
var
    i : integer;
begin
    ambil.Enabled:=false;
    temp6:=random(data_simpan-80);
    for i:=0 to 79 do //menampilkan cuplikan awal 80 chip
    begin
        str(data[0,i+(temp6+1)]:0:3,temp0);
        s:=temp0;
        if length(s) < 6 then s := ' ' + s;
        Memo5.Text:= s + ' ' + Memo5.Text;

        str(data[1,i+(temp6+1)]:0:3,temp0);
        s:=temp0;
        if length(s) < 6 then s := ' ' + s;
        Memo6.Text:= s + ' ' + Memo6.Text;
    end;

    labell11.Caption:= 'Informasi Cuplik : ' + inttostr(temp6) + ' - '
+ inttostr(temp6+79);

    for i:=0 to 15 do //menampilkan cuplikan 16 data yg dikorelasi
    begin
        str(data[0,i+(temp6+1)]:0:3,temp0);
        s:=temp0;
        if length(s) < 6 then s := ' ' + s;
        Memo7.Text:= s + ' ' + Memo7.Text;

        str(data[0,i+(temp6+65)]:0:3,temp0);
        s:=temp0;
        if length(s) < 6 then s := ' ' + s;
        Memo8.Text:= s + ' ' + Memo8.Text;
    end;
end;

```

```

str(data[1,i+(temp6+1)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Memo9.Text:= s + ' ' + Memo9.Text;

str(data[1,i+(temp6+65)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Memo10.Text:= s + ' ' + Memo10.Text;
end;

jendela1;
end;

procedure TForm1.jendela2;
var
i,temp7,temp8 :integer;
hasil0,hasil1:string;
begin
memo11.Clear; memo12.Clear; memo13.Clear; memo14.Clear;
temp7:=strtoint(label3.Caption);
temp7:=temp7+1;
temp8:=temp7+16;
for i:=0 to 79 do
begin
str(data[0,i+(temp6+temp7)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Memo11.Text:= s + ' ' + Memo11.Text;

str(data[1,i+(temp6+temp7)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Memo12.Text:= s + ' ' + Memo12.Text;
end;

label3.Caption:=inttostr(temp7);

for i:=0 to 15 do
begin
if edit8.Text >= '0' then stop.Click;
with StringGrid8 do
begin
str(data[0,i+(temp6+temp7)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[1,i+1]:= ' ' + s;

str(data[0,i+(temp6+temp7+64)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[2,i+1]:= ' ' + s;

data1[0,i]:=data[0,i+(temp6+temp7)]*data[0,i+(temp6+temp7+64
)];
str(data1[0,i]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[3,i+1]:= ' ' + s;

```

```

kuadrat[0,i]:=power(data[0,i+(temp6+temp7+64)],2);
str(kuadrat[0,i]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[4,i+1]:= ' ' + s;
end;

with StringGrid9 do
begin
str(data[1,i+(temp6+temp7)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[1,i+1]:= ' ' + s;

str(data[1,i+(temp6+temp7+64)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[2,i+1]:= ' ' + s;

data1[1,i]:=data[1,i+(temp6+temp7)]*data[1,i+(temp6+temp7+64)];
str(data1[1,i]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[3,i+1]:= ' ' + s;

kuadrat[1,i]:=power(data[1,i+(temp6+temp7+64)],2);
str(kuadrat[1,i]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
cells[4,i+1]:= ' ' + s;
end;
end;
hitung0[0,0]:=data1[0,0]+data1[0,1]+data1[0,2]+data1[0,3];
hitung0[0,1]:=data1[0,4]+data1[0,5]+data1[0,6]+data1[0,7];
hitung0[0,2]:=data1[0,8]+data1[0,9]+data1[0,10]+data1[0,11];
hitung0[0,3]:=data1[0,12]+data1[0,13]+data1[0,14]+data1[0,15];

str((hitung0[0,0]+hitung0[0,1]+hitung0[0,2]+hitung0[0,3]):0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
edit2.Text:= ' ' + s;
edit6.Text:= ' ' + s;

hitung0[1,0]:=data1[1,0]+data1[1,1]+data1[1,2]+data1[1,3];
hitung0[1,1]:=data1[1,4]+data1[1,5]+data1[1,6]+data1[1,7];
hitung0[1,2]:=data1[1,8]+data1[1,9]+data1[1,10]+data1[1,11];
hitung0[1,3]:=data1[1,12]+data1[1,13]+data1[1,14]+data1[1,15];

str((hitung0[1,0]+hitung0[1,1]+hitung0[1,2]+hitung0[1,3]):0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
edit3.Text:= ' ' + s;
edit9.Text:= ' ' + s;

hitung0[2,0]:=kuadrat[0,0]+kuadrat[0,1]+kuadrat[0,2]+kuadrat[0,3];
hitung0[2,1]:=kuadrat[0,4]+kuadrat[0,5]+kuadrat[0,6]+kuadrat[0,7];
hitung0[2,2]:=kuadrat[0,8]+kuadrat[0,9]+kuadrat[0,10]+kuadrat[0,11];
hitung0[2,3]:=kuadrat[0,12]+kuadrat[0,13]+kuadrat[0,14]+kuadrat[0,15];
str((hitung0[2,0]+hitung0[2,1]+hitung0[2,2]+hitung0[2,3]):0:3,temp0);

```

```

s:=temp0;
if length(s) < 6 then s := ' ' + s;
edit4.Text:= ' ' + s;
edit7.Text:= ' ' + s;
hitung0[3,0]:=kuadrat[1,0]+kuadrat[1,1]+kuadrat[1,2]+kuadrat[1,3];
hitung0[3,1]:=kuadrat[1,4]+kuadrat[1,5]+kuadrat[1,6]+kuadrat[1,7];
hitung0[3,2]:=kuadrat[1,8]+kuadrat[1,9]+kuadrat[1,10]+kuadrat[1,11];
hitung0[3,3]:=kuadrat[1,12]+kuadrat[1,13]+kuadrat[1,14]+kuadrat[1,15];
str((hitung0[3,0]+hitung0[3,1]+hitung0[3,2]+hitung0[3,3]):0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
edit5.Text:= ' ' + s;
edit10.Text:= ' ' + s;
temp10[0,0]:=hitung0[0,0]+hitung0[0,1]+hitung0[0,2]+hitung0[0,3];
temp10[0,1]:=hitung0[2,0]+hitung0[2,1]+hitung0[2,2]+hitung0[2,3];
str((temp10[0,0]/temp10[0,1]):0:3,temp0);
hasil0:=temp0;
edit8.Text:= ' ' + hasil0;

temp10[0,2]:=hitung0[1,0]+hitung0[1,1]+hitung0[1,2]+hitung0[1,3];
temp10[0,3]:=hitung0[3,0]+hitung0[3,1]+hitung0[3,2]+hitung0[3,3];
str((temp10[0,2]/temp10[0,3]):0:3,temp0);
hasil1:=temp0;
edit11.Text:= ' ' + hasil1;

if hasil1 >= ComboBox3.Text then
begin
if hasil0 >= ComboBox3.Text then
begin
edit11.Text:= ' ' + hasil1;
edit11.Color:=clLime;
Berhenti:=true;
start.Enabled:=true;
button1.Enabled:=true;
stop.Enabled:=false;
ComboBox1.Enabled:=true;
ComboBox1.Color:=clWindow;
Edit1.Enabled:=true;
Edit1.Color:=clWindow;
edit8.Text:= ' ' + hasil0;
edit8.Color:=clLime;
Edit1.Enabled:=true;
Edit1.Color:=clWindow;
end;

for i:=0 to 63 do //menampilkan hasil jendela FFT
begin
str(data[0,i+(temp6+temp8)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Mem013.Text:= s + ' ' + Mem013.Text;

str(data[1,i+(temp6+temp8)]:0:3,temp0);
s:=temp0;
if length(s) < 6 then s := ' ' + s;
Mem014.Text:= s + ' ' + Mem014.Text;
end;

end ;
end;

```

```
procedure TForm1.jendela1;  
begin  
    jendela:=true;  
end;  
  
procedure TForm1.StopClick(Sender: TObject);  
begin  
    Berhenti:=true;  
    start.Enabled:=true;  
    stop.Enabled:=false;  
    button1.Enabled:=true;  
    ambil.Enabled:=false;  
end;  
  
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    if MessageDlg('Anda yakin ingin keluar  
?',mtConfirmation,[mbYes]+[mbNo],0)=mrYes then application.Terminate;  
end;  
  
end.
```



