



TESIS - EE185401

# PENGEMBANGAN DAN EVALUASI SISTEM KOMUNIKASI DIGITAL HF BERBASIS SOFTWARE-DEFINED RADIO

AGUNG WICAKSONO  
07111850030002

DOSEN PEMBIMBING  
Prof. Ir. Gamantyo Hendranto, M.Eng., Ph.D.  
Dr. Ir. Achmad Mauludiyanto, M.T.

PROGRAM MAGISTER  
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2020





TESIS - EE185401

**PENGEMBANGAN DAN EVALUASI SISTEM  
KOMUNIKASI DIGITAL HF BERBASIS  
SOFTWARE-DEFINED RADIO**

AGUNG WICAKSONO  
07111850030002

DOSEN PEMBIMBING  
Prof. Ir. Gamantyo Hendranto, M.Eng., Ph.D.  
Dr. Ir. Achmad Mauludiyanto, M.T.

PROGRAM MAGISTER  
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2020



## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

**Magister Teknik (MT)**

di

**Institut Teknologi Sepuluh Nopember**

Oleh:

**AGUNG WICAKSONO**

**NRP: 07111850030002**

Tanggal Ujian: 9 Januari 2020

Periode Wisuda: Maret 2020

Disetujui oleh:

**Pembimbing:**

1. Prof. Ir. Gamantyo Hendranto,  
M.Eng.,Ph.D.  
NIP: 197011111993031002

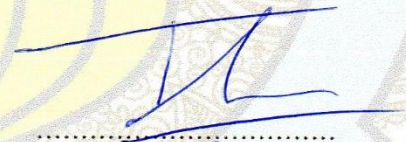


2. Dr. Ir. Achmad Mauludiyanto, M.T.  
NIP: 196109031989031001

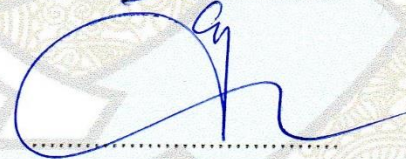


**Penguji:**

1. Dr. Istas Pratomo, S.T., M.T.  
NIP: 197903252003121001



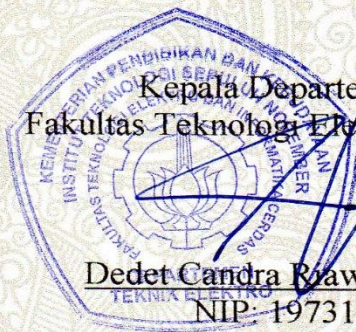
2. Eko Setijadi, S.T., M.T., Ph.D.  
NIP: 197210012003121002



3. Dr. Ir. Wirawan, DEA  
NIP: 196311091989031011



Kepala Departemen Teknik Elektro  
Fakultas Teknologi Elektro dan Informatika Cerdas



Dedet Candra Wirawan, S.T., M.Eng., Ph.D.  
NIP: 197311192000031001

*Halaman ini sengaja dikosongkan*

## PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**PENGEMBANGAN DAN EVALUASI SISTEM KOMUNIKASI DIGITAL HF BERBASIS SOFTWARE-DEFINED RADIO**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 19 Desember 2019



Agung Wicaksono

NRP. 07111850030002

# **PENGEMBANGAN DAN EVALUASI SISTEM KOMUNIKASI DIGITAL HF BERBASIS SOFTWARE-DEFINED RADIO**

Nama mahasiswa : Agung Wicaksono  
NRP : 07111850030002  
Pembimbing : 1. Prof. Ir. Gamantyo Hendrantoro, M.Eng., Ph.D.  
2. Dr. Ir. Achmad Mauludiyanto, M.T.

## **ABSTRAK**

Sistem komunikasi High Frequency (HF) bekerja menggunakan gelombang radio pada frekuensi 3-30 MHz yang merambat dalam bentuk skywave dengan bantuan lapisan ionosfer. Untuk aplikasi sistem komunikasi darurat, sistem komunikasi HF diharapkan dapat mengirimkan pesan berupa teks dengan menerapkan sistem komunikasi digital. Makalah ini melaporkan desain sistem komunikasi digital HF menggunakan perangkat Software-Defined Radio (SDR) untuk implementasi yang mudah. Agar dapat dicapai error rate performance yang tinggi, digunakan protokol AX.25 pada data link layer. Evaluasi sistem dilakukan dengan eksperimen untuk mengukur Signal To Noise Ratio (SNR), Bit Error Rate (BER), Character Error Rate (CER) dan Frame Error Rate (FER) pada sistem tanpa dan dengan protokol AX.25. Hasil eksperimen menunjukkan bahwa implementasi sistem pada platform USRP dengan LabView sebagai user interface berhasil dilakukan dan bahwa penggunaan protokol AX.25 menghasilkan zero error rate.

Kata kunci: Error Rate, Protokol AX.25, Sistem Komunikasi Digital HF, Software-Defined Radio.



*Halaman ini sengaja dikosongkan*

# **DEVELOPMENT AND EVALUATION OF AN HF DIGITAL COMMUNICATION SYSTEM BASED ON SOFTWARE-DEFINED RADIO**

By : Agung Wicaksono  
Student Identity Number : 07111850030002  
Supervisors : 1. Prof. Ir. Gamantyo Hendrantoro,  
M.Eng.,Ph.D.  
2. Dr. Ir. Achmad Mauludiyanto, M.T.

## **ABSTRACT**

High Frequency (HF) communication systems use radio waves at frequencies in the range of 3-30 MHz propagating in the form of skywave with the help of the ionosphere. For emergency communication applications, the HF communication system is expected to send text messages by implementing a digital communication system. This paper presents the design of the HF digital communication system on a Software Defined Radio (SDR) platform for easy implementation. To achieve a good error rate performance, the AX.25 protocol is used at the data link layer. The system performance is evaluated by measuring Signal to Noise Ratio (SNR), Bit Error Rate (BER), Character Error Rate (CER), and Frame Error Rate (FER) with and without the AX.25 protocol. The results show that the system has been successfully implemented on the USRP platform with LabView as the user interface, while the use of the AX.25 protocol generates zero error rates.

Key words: Error Rate, AX.25 Protocol, HF Digital Communication System, Software-Defined Radio.

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyusun dan menyelesaikan Laporan Tesis dengan judul **“PENGEMBANGAN DAN EVALUASI SISTEM KOMUNIKASI DIGITAL HF BERBASIS SOFTWARE-DEFINED RADIO”**. Laporan Tesis ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Magister Teknik pada Program Magister Bidang Keahlian Telekomunikasi Multimedia Departemen Teknik Elektro Institut Teknologi Sepuluh Nopember.

Dalam melakukan penelitian dan penyusunan Laporan Tesis ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terimakasih yang tak terhingga kepada :

1. Drs. Hidayaturohman, M.Pd., Kusyatmi, S.Pd. M.Pd., selaku kedua orang tua, Ardea Himawan Nugroho selaku adik penulis, Bella Allivia Sativa Turmono, S.Pd., selaku teman perjalanan kehidupan penulis, Keluarga Besar KH. Abdus Salam dan Keluarga Besar Ranawijaya, yang tidak henti-hentinya memberikan doa dan dukungan untuk menyelesaikan laporan tesis.
2. Bapak Prof. Ir. Gamantyo Hendratoro, M.Eng., Ph.D., selaku dosen wali, dosen pembimbing dan dosen mata kuliah yang telah memberikan bimbingan, arahan, dan waktu sehingga penulis dapat menyelesaikan laporan tesis.
3. Bapak Dr. Ir. Achmad Mauludiyanto, M.T., selaku dosen pembimbing dan dosen mata kuliah yang telah memberikan bimbingan, arahan, dan waktu sehingga penulis dapat menyelesaikan laporan tesis.
4. Ibu Dr. Umairah, S.Tr., selaku rekan tim HF dalam penelitian yang telah memberikan arahan, pengalaman dan ilmu selama melakukan penelitian sehingga penulis dapat menyelesaikan laporan tesis.
5. Ibu Dr. Indah Kurniawati, S.T., M.T., selaku rekan tim HF dalam penelitian yang telah memberikan arahan, pengalaman, dan ilmu selama melakukan penelitian sehingga penulis dapat menyelesaikan laporan tesis.

6. Bapak Varuliantor Dear, S.T., M.T., Pusat Sains Antariksa Lembaga Penerbangan dan Antariksa Nasional (LAPAN) selaku rekan yang telah memberikan peminjaman perangkat penelitian sehingga penulis dapat menyelesaikan laporan tesis.
7. Bapak Wahyu Pamungkas, S.T., M.T., mahasiswa S-3 Teknik Elektro Bidang Keahlian Telekomunikasi Multimedia selaku rekan diskusi dan rekan berbagi penulis di Surabaya yang telah memberikan pengalaman, arahan, bimbingan, dan pengajaran sehingga penulis dapat menyelesaikan laporan tesis.
8. Bapak Teguh Imam Suharto, S.T., M.T., selaku rekan tim HF dalam penelitian yang telah membantu dalam proses penelitian sehingga penulis dapat menyelesaikan laporan tesis.
9. Bapak Irawan Budiadi Luhur, S.T., Bapak Geordano Vincenzo, S.T., Bapak Afrizal Noor Karim, S.T., Ibu Vina Amalia, S.T., dan Ibu Saraah Nabilah Darmawan, S.T., selaku rekan tim HF dalam penelitian yang telah membantu dalam proses penelitian sehingga penulis dapat menyelesaikan laporan tesis.
10. Kementrian Riset, Teknologi dan Pendidikan Tinggi (RISTEKDIKTI) yang telah memberikan hibah dana penelitian dengan skema Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) sehingga penulis dapat menyelesaikan penelitian dan laporan tesis.
11. Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) ITS yang telah memberikan hibah dana penelitian dengan skema Lab Based Education (LBE) sehingga penulis dapat menyelesaikan penelitian dan laporan tesis.
12. Seluruh dosen program Pascasarjana Teknik Elektro khususnya dosen bidang keahlian Telekomunikasi Multimedia yang telah memberikan arahan dan bimbingan untuk mendalami ilmu Telekomunikasi.
13. Keluarga Besar Laboratorium Antena dan Propagasi ITS khususnya untuk Ibu Devy Kuswidiastuti, S.T., M.Sc., Bapak Syafrizal Tachfulloh, S.T., M.T., Bapak Fannush Akbar, S.Tr., Ibu Fitri Amalia, S.T., M.T., dan Ibu Ainnur Rahayu, S.ST. yang telah memberikan dukungan sehingga penulis dapat menyelesaikan laporan tesis.
14. Rekan-rekan mahasiswa/i S-1 dan S-2 ada Pak Wildan Bramaji Wahyudianto, S.T., Pak Zainal Abidin, S.Kom., Pak M. Arifin, S.T., dan Ibu Windi

Puspitasari, S.ST., selaku teman berbagi dan kerjasama selama perkuliahan sehingga penulis dapat menyelesaikan laporan tesis.

15. Rekan-rekan mahasiswa/i S-1, S-2, dan S-3 Teknik Elektro Bidang Keahlian Telekomunikasi Multimedia di Laboratorium B.301, B.304 dan B.306 yang telah memberikan dukungan.
16. Kepada semua pihak yang telah membantu dan tidak dapat penulis sebutkan satu persatu.

Surabaya, 19 Desember 2019

Penulis

Agung Wicaksono

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

LEMBAR PENGESAHAN TESIS .....	iii
PERNYATAAN KEASLIAN TESIS .....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xxi
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan .....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah .....	3
1.5 Kontribusi .....	4
BAB 2 KAJIAN PUSTAKA.....	5
2.1 Sistem Komunikasi Radio HF .....	5
2.1.1 Mode Propagasi .....	5
2.1.2 Skip Zone.....	8
2.1.3 Teknik <i>Near Vertical Incidence Skywave</i> (NVIS).....	8
2.1.4 Fading .....	10
2.1.5 Frekuensi Kerja.....	11
2.1.6 Polarisasi.....	12
2.1.7 Link Budget .....	13
2.2 Sistem Komunikasi Digital.....	16
2.2.1 <i>Low Pass Equivalent</i> .....	18
2.2.2 Modulasi <i>Binary Phase-Shift Keying</i> (BPSK).....	20
2.2.3 <i>Signal to Noise Ratio</i> (SNR).....	22
2.2.4 Kinerja <i>Error Rate</i> .....	22



2.2.5 <i>Filter Raised Cosine</i> .....	23
2.2.6 <i>Low Pass Filter</i> .....	24
2.2.7 Sinkronisasi Gelombang Pembawa .....	25
2.2.8 <i>Protocol AX.25</i> .....	28
2.2.9 Sinkronisasi Data Pada Protocol AX.25 .....	36
2.3 <i>Software-Defined Radio</i> .....	40
2.4 Antena Dipole.....	43
<b>BAB 3 METODE PENELITIAN</b> .....	47
3.1 Skema Penelitian .....	47
3.2 Diagram Alir Penelitian.....	48
3.3 Rancangan Penelitian.....	50
3.3.1 Desain Frekuensi Kerja.....	50
3.3.2 Desain Lokasi Pemancar dan Penerima.....	50
3.4 Desain Sistem Keseluruhan .....	51
3.4.1 Desain Sistem Pemancar.....	54
3.4.2 Desain Sistem Penerima .....	64
3.5 Desain Parameter .....	74
3.5.1 Desain Parameter di Pemancar .....	74
3.5.2 Desain Parameter di Penerima.....	75
3.6 Desain Link Budget .....	76
3.7 Indikator Kinerja.....	79
3.7.1 <i>Signal to Noise Ratio (SNR)</i> .....	79
3.7.2 <i>Bit Error Rate (BER)</i> .....	79
3.7.3 <i>Character Error Rate (CER)</i> .....	79
3.7.4 <i>Frame Error Rate (FER)</i> .....	80
<b>BAB 4 HASIL DAN PEMBAHASAN</b> .....	81
4.1 Hasil Implementasi Sistem Keseluruhan .....	81
4.1.1 Hasil Implementasi Sistem Pemancar Tanpa <i>Protocol AX.25</i> .....	85
4.1.2 Hasil Implementasi Sistem Pemancar Dengan <i>Protocol AX.25</i> ..	88
4.1.3 Hasil Implementasi Sistem Penerima Tanpa <i>Protocol AX.25</i> .....	91

4.1.4 Hasil Implementasi Sistem Penerima Dengan <i>Protocol AX.25</i> ...	93
4.2 Analisis Sitem Pemancar .....	96
4.2.1 Pembangkitan Sinyal Pada Pemancar Terhadap Frekuensi .....	96
4.2.2 Output Power Amplifier .....	97
4.2.3 Pengiriman Data Teks Berdasarkan Penggunaan Tanpa Protocol dan Dengan <i>Protocol AX.25</i> .....	98
4.3 Analisis Sistem Penerima .....	98
4.3.1 Analisis Hasil <i>Signal to Noise Ratio</i> (SNR) Rata-Rata Berdasarkan Penggunaan Tanpa Protcol dan Dengan Protocol <i>AX.25</i> .....	98
4.3.2 Analisis Hasil <i>Bit Error Rate</i> (BER) Rata-Rata Berdasarkan Penggunaan Tanpa Protcol dan Dengan <i>Protocol AX.25</i> .....	99
4.3.3 Analisis Hasil <i>Character Error Rate</i> (CER) Rata-Rata Berdasarkan Penggunaan Tanpa Protcol dan Dengan <i>Protocol AX.25</i> .....	99
4.3.4 Analisis Hasil <i>Frame Error Rate</i> (FER) Rata-Rata Berdasarkan Penggunaan Tanpa Protcol dan Dengan <i>Protocol AX.25</i> .....	100
4.4 Sintesis.....	101
BAB 5 KESIMPULAN.....	103
5.1 KESIMPULAN .....	103
5.2 SARAN.....	105
DAFTAR PUSTAKA .....	107
LAMPIRAN 1.....	111
LAMPIRAN 2.....	153
LAMPIRAN 3.....	165
LAMPIRAN 4.....	166
LAMPIRAN 5.....	167
LAMPIRAN 6.....	173
DAFTAR RIWAYAT HIDUP.....	174

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2. 1 Propagasi Gelombang Radio HF.....	5
Gambar 2. 2 Lapisan Ionosfer.....	6
Gambar 2. 3 Mode Propagasi Gelombang Sederhana .....	7
Gambar 2. 4 Mode Propagasi Gelombang Kompleks.....	7
Gambar 2. 5 Daerah Skip Zone.....	8
Gambar 2. 6 Sudut Elevasi Teknik NVIS .....	9
Gambar 2. 7 Propagasi Sinyal dengan Teknik NVIS.....	9
Gambar 2.8 Ketidakfokusan Lintasan Disebabkan Oleh Fading .....	11
Gambar 2.9 Komponen sistem komunikasi digital .....	18
Gambar 2.10 Blok Diagram Low Pass Equivalent .....	19
Gambar 2.11 BPSK <i>Modulator</i> dan <i>Waveform</i> .....	20
Gambar 2.12 BPSK <i>Constellation</i> .....	21
Gambar 2.13 BPSK <i>Demodulator</i> .....	21
Gambar 2.14 <i>Filter Raised-Cosine</i> Dengan Ragam <i>Roll-Of Factor</i> .....	24
Gambar 2.15 Filter LPF Ideal .....	25
Gambar 2.16 Multipath Effect dan Doppler Shift.....	26
Gambar 2.17 Skema Dasar PLL.....	26
Gambar 2.18 Proses Kondisi Steady State Sistem PLL.....	28
Gambar 2.19 Ilustrasi Struktur Protokol Ax.25 .....	29
Gambar 2.20 Format Frame .....	36
Gambar 2.21 Blok Diagram SDR .....	41
Gambar 2.22 Struktur Bagan USRP N210.....	42
Gambar 2.23 Antena Dipole $\frac{1}{2}$ lamda.....	45
Gambar 3.1 Skema Penelitian (fishbone).....	47
Gambar 3.2 Diagram Alir Penelitian .....	49
Gambar 3.2 Antena Pemancar (TX) di Jembatan Gedung AJ .....	51
Gambar 3.3 Antena Penerima (RX) di Gedung B.....	51
Gambar 3.4 Desain Sistem Komunikasi Digital HF Keseluruhan.....	53

Gambar 3.4 Desain Sistem Pemancar (TX) .....	54
Gambar 3.5 Desain Sistem Tanpa Protocol di Sisi Pemancar .....	55
Gambar 3.6 Tampilan Sub-Sub Block Sistem Modulasi .....	56
Gambar 3.7 Desain Gray Code Di Dalam Sub System.....	56
Gambar 3.8 Desain sub sistem Packet Generate .....	57
Gambar 3.9 Desain Modulasi.....	58
Gambar 3.10 Desain Sistem Dengan Protocol di Sisi Pemancar .....	58
Gambar 3.11 Desain Frame Protocol AX.25 .....	59
Gambar 3.12 Desain input teks pada software labview .....	60
Gambar 3.13 Desain tampilan encoder menggunakan protocol AX.25.....	60
Gambar 3.14 Desain Penerima (RX) .....	65
Gambar 3.15 Modul MT Phase-Locked Loop .....	66
Gambar 3.16 Desain Blok Diagram Modul PLL .....	67
Gambar 3.17 Desain Sistem Tanpa Protocol di Sisi Penerima .....	68
Gambar 3.18 Integrasi Desain PLL dengan Sistem Tanpa Protocol di Sisi Penerima.....	69
Gambar 3.19 Desain Sistem Dengan Protocol di Sisi Penerima.....	70
Gambar 3.20 Desain Noise Pada Software Labview .....	70
Gambar 3.21 Mathscript node pada labview.....	73
Gambar 3.22 Desain Output.....	74
Gambar 3.23 Tampilan Input Parameter di Pemancar Pada Software Labview .....	75
Gambar 3.24 Tampilan Input Parameter di Penerima Pada Software Labview .....	76
Gambar 4.1 Konfigurasi Perangkat Pemancar .....	81
Gambar 4.2 Blok Diagram Sistem Pengukuran di Pemancar .....	82
Gambar 4.3 Konfigurasi Perangkat Penerima.....	83
Gambar 4.4 Blok Diagram Sistem Pengukuran di Penerima.....	84
Gambar 4.5 Implementasi Pengiriman Teks 500 Karakter Tanpa Protocol	85
Gambar 4.6 Implementasi Pengiriman Teks 700 Karakter Tanpa Protocol	86

Gambar 4.7 Implementasi Pengiriman Teks 1000 Karakter Tanpa Protocol	86
Gambar 4.8 Implementasi Pengiriman Teks 1500 Karakter Tanpa Protocol	87
Gambar 4.9 Implementasi Pengiriman Teks 2000 Karakter Tanpa Protocol	87
Gambar 4.10 Implementasi Pengiriman Teks 500 Karakter Dengan Protocol	88
Gambar 4.11 Implementasi Pengiriman Teks 700 Karakter Dengan Protocol	89
Gambar 4.12 Implementasi Pengiriman Teks 1000 Karakter Dengan Protocol .....	89
Gambar 4.13 Implementasi Pengiriman Teks 1500 Karakter Dengan Protocol .....	90
Gambar 4.14 Implementasi Pengiriman Teks 2000 Karakter Dengan Protocol .....	90
Gambar 4.15 Salah Satu Implementasi Penerimaan Teks 500 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa Protocol .....	91
Gambar 4.16 Salah Satu Implementasi Penerimaan Teks 700 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa Protocol .....	92
Gambar 4.17 Salah Satu Implementasi Penerimaan Teks 1000 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa Protocol .....	92
Gambar 4.18 Salah Satu Implementasi Penerimaan Teks 1500 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa Protocol .....	92
Gambar 4.19 Salah Satu Implementasi Penerimaan Teks 2000 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa Protocol .....	93
Gambar 4.20 Salah Satu Implementasi Penerimaan Teks 500 Karakter Yang Berhasil Diterima Lengkap Dengan Protocol .....	94
Gambar 4.21 Salah Satu Implementasi Penerimaan Teks 700 Karakter Yang Berhasil Diterima Lengkap Dengan Protocol AX.25 .....	94
Gambar 4.22 Salah Satu Tampilan Hasil Penerimaan Teks 1000 Karakter Yang Berhasil Diterima Lengkap Dengan Penggunaan Protocol AX.25 .....	95
Gambar 4.23 Salah Satu Implementasi Penerimaan Teks 1500 Karakter Yang Berhasil Diterima Lengkap Dengan Protocol .....	95
Gambar 4.24 Salah Satu Implementasi Penerimaan Teks 2000 Karakter Yang Berhasil Diterima Lengkap Dengan Protocol .....	96
Gambar 4.25 Hasil Pembangkitan Sinyal Frekuensi Kerja.....	97
Gambar 4.26 Daya Transmitter.....	97

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2.1 Jenis redaman diantara transmitter dan receiver antenna lintasan HF .....	13
Tabel 2.2 Bentuk Frame U dan S .....	30
Tabel 2.3 Bentuk Frame Informasi .....	30
Tabel 2.4 Address Subfield .....	31
Tabel 2.5 Address Field .....	31
Tabel 2.6 Address Field of Frame on .....	32
Tabel 2.7 Frame AX.25 mode non-repeater .....	32
Tabel 2.8 Mode Repeater Address Field .....	33
Tabel 2.9 Frame Protokol AX.25 Dengan Satu Alamat Repeater .....	34
Tabel 2.10 Bentuk Control Field (Modulo 8) .....	35
Tabel 2.11 Bentuk Control Field (Modulo 128) .....	35
Tabel 2.12 Pembagian frekuensi untuk daughterboard .....	42
Tabel 3.1 Desain Isi Subfield .....	59
Tabel 3.2 Desain Parameter Sistem Komunikasi Digital HF pada sisi Pemancar .....	75
Tabel 3.3 Desain Parameter Sistem Komunikasi Digital HF pada sisi Pemancar .....	76
Tabel 3.3 Desain perhitungan skywave pathloss dengan menggunakan asumsi redaman tinggi .....	77
Tabel 3.4 Desain Perhitungan Link Budget .....	78
Tabel 3.4 Desain Perhitungan Link Budget (lanjutan) .....	79
Tabel 4.1 Parameter Variasi Jumlah Karakter Pengiriman .....	98
Tabel 4.2 Hasil Nilai SNR Rata-Rata Penggunaan Tanpa Protocol dan Dengan Protocol AX.25 .....	98
Tabel 4.3 . Hasil BER Rata-Rata Penggunaan Tanpa Protokol AX.25 dan Dengan Protokol AX.25 .....	99
Tabel 4.4 . Hasil CER Rata-Rata Penggunaan Tanpa Protocol AX.25 dan Dengan Protocol AX.25 .....	100



Tabel 4.5 Hasil FER Rata-Rata Penggunaan Tanpa Protocol AX.25 dan Dengan Protocol AX.25 .....	100
---	-----

*Halaman ini sengaja dikosongkan*



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Sistem komunikasi High Frequency (HF) menggunakan gelombang radio pada frekuensi 3 – 30 MHz, telah lama disukai karena kemudahan pengoperasian dan kemampuannya untuk menjangkau tempat-tempat yang jauhnya ribuan kilometer dengan memanfaatkan bantuan lapisan ionosfer<sup>[1]</sup>. Polarisasi gelombang radio yang merambat di ionosfer umumnya berbentuk bulat panjang, yang memiliki karakteristik pada titik keluar dan dapat di perkirakan dalam pendekatan deterministik sesuai dengan kondisi, selain itu terdapat dua polarisasi yang terkait diantaranya mode O dan X yang bersifat orthogonal<sup>[2]</sup>. *Near Vertical Incidence Skywave* (NVIS) adalah pemantulan satu kali lapisan F ionosfer, dengan sudut pancaran mendekati 90° atau vertikal, sehingga sinyal yang di pantulkan oleh lapisan ionosfer akan jatuh kembali ke area yang berjarak  $\pm 300$  km dari pemancar<sup>[3]</sup>. Dalam sistem komunikasi HF terdapat *fading* yang disebabkan oleh beberapa *multiple paths* dengan sinyal yang bervariasi dalam amplitude dan fase, tergantung pada waktu propagasi dan *bandwidth* dari sinyal yang di transmisikan<sup>[4]</sup>. Sistem komunikasi High Frequency (HF) telah diusulkan sebagai salah satu teknologi alternatif yang digunakan untuk teknologi telekomunikasi darurat atau diimplementasikan untuk daerah terisoalasi. Sistem komunikasi ini menawarkan infrastruktur dengan biaya rendah dan teknologi ketahanan tinggi dibandingkan dengan komunikasi satelit<sup>[5]</sup>.

Sistem komunikasi digital HF dapat berjalan dengan baik di sebuah sistem dengan menggunakan perangkat keras *Software-Defined Radio* (SDR). SDR adalah perangkat sistem komunikasi radio dimana di dalamnya terdapat beberapa komponen diantaranya *mixer*, *filter*, *modulator*, *demodulator*, *detektor* dan *Phase-Locked Loop* (PLL) yang dapat di implementasikan ke berbagai perangkat lunak diantaranya GNU

Radio, *LabView* dan *LabView Communication*<sup>[6]</sup>. *LabView* merupakan perangkat lunak yang dimaksud untuk menjalankan sistem komunikasi digital HF pada perangkat SDR. *LabView* adalah bahasa pemrograman grafis yang menggunakan pemrograman data *flow*, diagram Blok dan *instrument virtual* dalam perencanaan dan pengembangan sistem komunikasi digital HF<sup>[7]</sup>. Pada jaringan radio paket, protokol data *link* AX.25 paling sering digunakan. Protokol AX.25 adalah *layer* data *link* yang digunakan hampir semua radio digital dan protokolnya telah digunakan dalam sistem komunikasi HF, VHF, UHF dan sistem komunikasi satelit<sup>[8]</sup>. Protokol AX.25 merupakan varian dari protokol internasional X.25 yang diterapkan pada sistem komunikasi nirkabel dan sistem komunikasi radio paket<sup>[9]</sup>. Namun, sistem ini menyajikan *bandwidth* koherensi terbatas, biasanya di perkirakan beberapa kilohertz yang muncul sebagai faktor pembatas utama dalam desain sistem komunikasi digital. Kinerja peralatan yang tersedia dalam hal kecepatan transfer data moderat untuk HF bervariasi dari 4,8 – 9,9 kbps<sup>[10]</sup> dalam 10 kHz *bandwidth*.

Berdasarkan uraian tersebut maka pada penelitian ini melaporkan tentang desain, implementasi dan uji coba sistem komunikasi HF digital dengan protokol AX.25 pada *platform* SDR.

## 1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini antara lain sebagai berikut:

1. Bagaimana desain sistem komunikasi digital HF berbasis *Software-Defined Radio*?
2. Bagaimana implementasi sistem komunikasi digital HF berbasis *Software-Defined Radio*?
3. Bagaimana uji coba penggunaan dan tanpa penggunaan protokol AX.25 terhadap kinerja sistem komunikasi digital HF berbasis *Software-Defined Radio*?

### 1.3 Tujuan

Tujuan penelitian ini adalah untuk mengetahui desain, implementasi, dan ujicoba penggunaan protokol dan tanpa protokol AX.25 pada sistem komunikasi digital HF. Indikator kinerja yang dimaksud berupa:

1. *Signal to Noise Ratio* (SNR)
2. *Bit Error Rate* (BER)
3. *Character Error Rate* (CER)
4. *Frame Error Rate* (FER)

### 1.4 Batasan Masalah

Batasan masalah pada penelitian ini antara lain sebagai berikut:

1. Perangkat *Software-Defined Radio* (SDR) yang digunakan pada sistem komunikasi digital HF yaitu USRP N210.
2. Perangkat lunak yang digunakan pada sistem komunikasi digital HF yaitu Labview 2014.
3. Propagasi yang digunakan pada sistem komunikasi digital HF yaitu *Near Vertical Incidence Sky Wave* (NVIS).
4. Frekuensi kerja di rencanakan yang akan digunakan pada pemancar dan penerima yaitu 7 MHz.
5. Uji coba sistem komunikasi digital HF ini memperhatikan kondisi ionosfer.
6. Antena yang digunakan pada uji coba sistem komunikasi digital HF yaitu antena *horizontal* dipole  $\lambda/2$  untuk sisi pemancar dan penerima.
7. Protokol yang digunakan pada sistem komunikasi digital HF yaitu protokol Ax.25.
8. Uji coba sistem komunikasi digital HF yaitu menggunakan modulasi BPSK.
9. Uji coba sistem komunikasi digital HF yaitu menggunakan *transmit filter root raised cosine* dengan nilai *roll of factor* 0,5.
10. Uji coba sistem komunikasi digital HF yaitu menggunakan *IQ Rate*

dengan nilai 200 Ksymbol/sec.

11. Uji coba sistem komunikasi digital HF yaitu menggunakan *samples per symbol* dengan nilai 30.
12. Uji coba sistem komunikasi digital HF menggunakan bandwidth 10 KHz sesuai Peraturan Organisasi Amatir Radio Indonesia (ORARI) Nomor: KEP-065/OP/KU/2009 dan Peraturan Menteri Komunikasi dan Informatika Indonesia No.33 Tahun 2009 tentang Penyelenggaraan Amatir Radio.
13. Uji coba sistem komunikasi digital HF menggunakan pengiriman teks dengan variasi pengiriman teks 500, 700, 1000, 1500 dan 2000 karakter dengan pengiriman data teks pengujian sebanyak 10 kali pada masing-masing variasi pengiriman.
14. Parameter yang diujikan pada uji coba sistem komunikasi digital HF yaitu *Signal to Noise Ratio* (SNR), *Bit Error Rate* (BER), *Character Error Rate* (CER), dan *Frame Error Rate* (FER).
15. Uji coba sistem komunikasi digital HF dilakukan dengan Antena di Jembatan Gedung AJ (sebagai pemancar) dan Antena di Gedung B (sebagai penerima) di kampus Institut Teknologi Sepuluh Nopember.

## 1.5 Kontribusi

Penelitian ini diharapkan dapat memberikan kontribusi sebagai berikut:

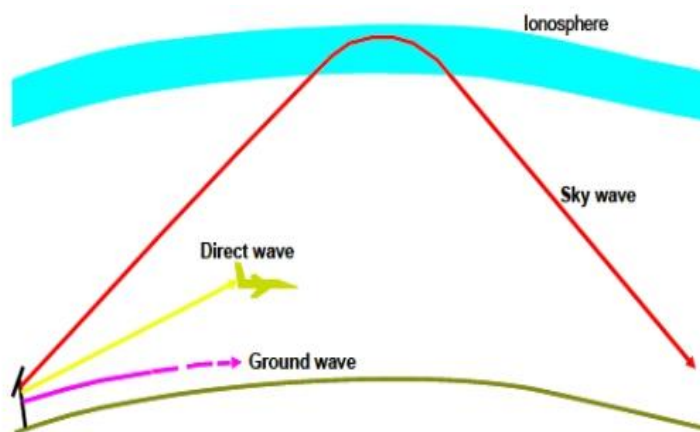
1. Memberikan informasi berupa desain sistem komunikasi digital HF berbasis *Software-Defined Radio*, untuk pengembangan sistem komunikasi digital HF jarak jauh.
2. Memberikan informasi berupa implementasi sistem komunikasi digital HF berbasis *Software-Defined Radio*.
3. Memberikan informasi berupa uji coba penggunaan dan tanpa penggunaan protokol AX.25 terhadap kinerja sistem komunikasi digital HF berbasis *Software-Defined Radio*.

## BAB 2

### KAJIAN PUSTAKA

#### 2.1 Sistem Komunikasi Radio HF

Komunikasi radio HF (High-Frequency) berdasarkan rekomendasi ITU memiliki frekuensi kerja yaitu 3-30 MHz. HF merupakan sistem komunikasi radio yang memanfaatkan lapisan ionosfer sebagai pemantul sinyal gelombangnya. Gelombang yang dipantulkan oleh lapisan ionosfer tersebut dinamakan gelombang ionosfer (ionospheric wave) atau gelombang langit (skywave). Gelombang dipantulkan oleh kumpulan partikel-partikel yang terionisasi di lapisan ionosfer seperti yang ditunjukkan pada Gambar 2.1. Gelombang yang diterima berfluktuasi sebanding dengan kondisi partikel yang ada di lapisan. Kondisi yang baik memungkinkan gelombang yang dikirimkan dapat dipantulkan kembali ke bumi pada jarak tertentu dengan kondisi gelombang yang tidak tembus ke luar angkasa<sup>[1]</sup>.



Gambar 2. 1 Propagasi Gelombang Radio HF <sup>[11]</sup>

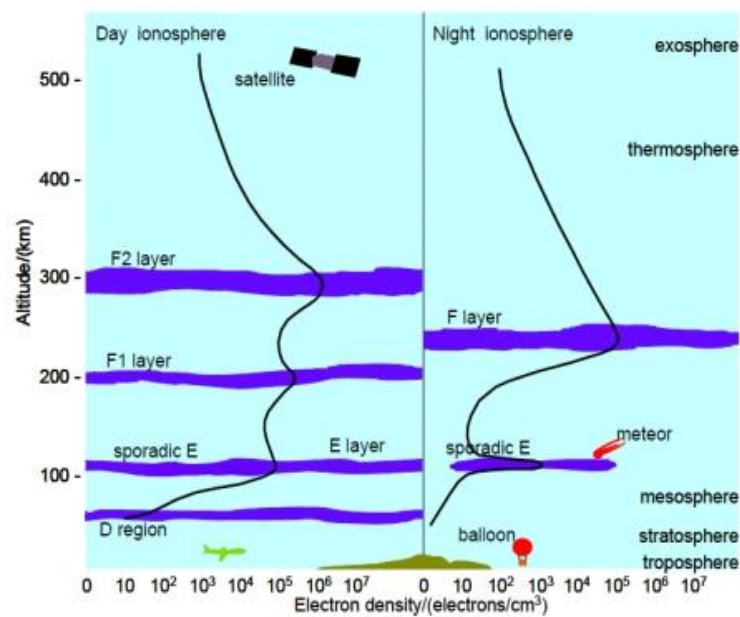
Ionosfer adalah lapisan pada atmosfer yang terletak pada ketinggian 50-500 km dari permukaan bumi. Lapisan ionosfer memiliki peran penting sebagai pemantul sinyal terdiri dari beberapa lapisan yaitu lapisan D, E, dan F. Masing-masing lapisan memiliki pengaruh bagi komunikasi radio HF. Gambar 2.2 menunjukkan bahwa lapisan F pada siang hari terbagi



menjadi dua lapisan yaitu F1 dan F2, kemudian pada malam hari lapisan F akan menjadi satu lapisan. Karena lapisan F ini selalu tersedia baik waktu siang maupun malam hari, maka diharapkan gelombang radio dapat berpropagasi pada lapisan ini.

Pada siang hari radiasi matahari akan mencapai nilai maksimum. Sebaliknya, pada malam hari akan mencapai minimum. Hal itu dapat mempengaruhi propagasi gelombang radio HF. Posisi matahari yang berubah-ubah terhadap titik-titik tertentu di bumi, dimana perubahan itu bisa terjadi harian, bulanan, dan tahunan. Tentunya perubahan posisi matahari tersebut akan mempengaruhi lapisan-lapisan ionosfer, sehingga karakteristiknya sulit untuk ditentukan<sup>[11]</sup>.

10



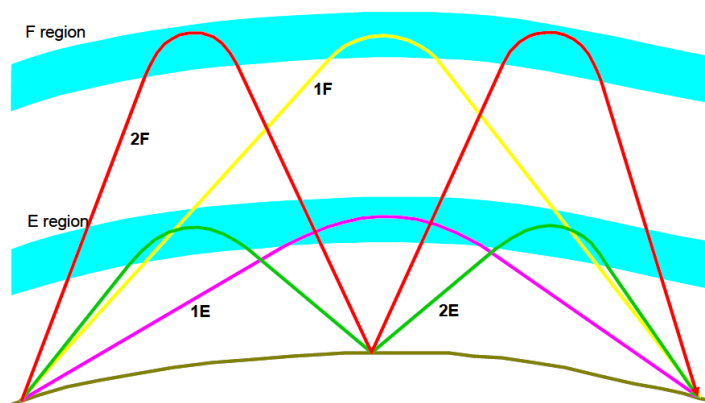
Gambar 2. 2 Lapisan Ionosfer <sup>[11]</sup>

### 2.1.1 Mode Propagasi

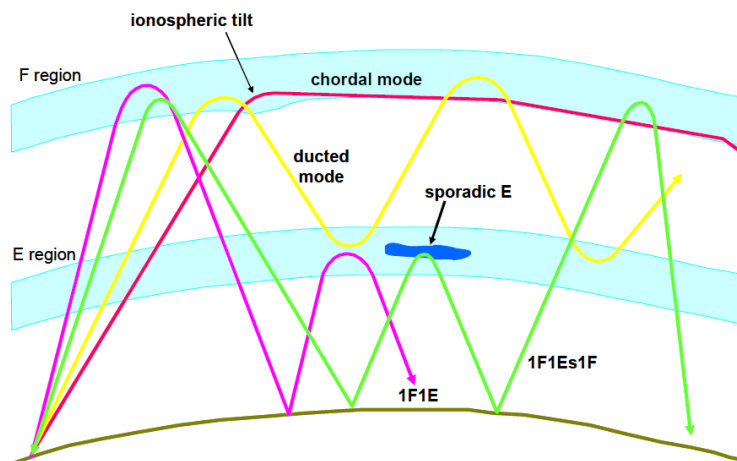
Mode propagasi merupakan lintasan atau jalur yang dilalui oleh gelombang radio untuk dapat berpropagasi dari pemancar ke penerima. Lintasan tersebut dapat banyak dan juga bervariasi bentuknya seperti ditunjukkan pada Gambar 2.3. Dimana gelombang radio yang keluar dari

sebuah pemancar akan langsung mengambil dan menempati salah satu bentuk lintasan untuk dapat berpropagasi.

Mode propagasi yang memiliki nomer hop paling sedikit disebut *first order mode*. Sedangkan mode yang membutuhkan hop tambahan biasa disebut *second order mode* dan begitu seterusnya. Contohnya ialah untuk sebuah lintasan dengan jarak 5000 km, maka akan memiliki *first order* pada mode F yaitu 2 buah hop (2F), kemudian memiliki *second order* pada mode F yaitu sebanyak 3 buah hop (3F) dan seterusnya.



Gambar 2. 3 Mode Propagasi Gelombang Sederhana <sup>[11]</sup>



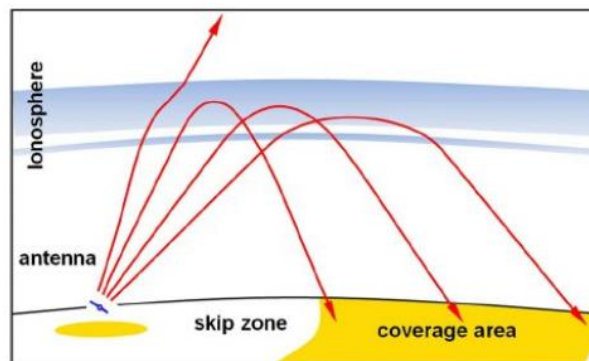
Gambar 2. 4 Mode Propagasi Gelombang Kompleks <sup>[10]</sup>

Mode propagasi gelombang sederhana seperti pada Gambar 2.3 menunjukkan bahwa lintasan yang dilalui adalah per-satu buah lapisan ionosfer atau dapat dikatakan bahwa tidak menggunakan kombinasi penggunaan lapisan ionosfer. Sedangkan mode propagasi gelombang kompleks pada Gambar 2.4 menunjukkan bahwa lintasan yang dilalui oleh

gelombang radio merupakan kombinasi dari beberapa lapisan ionosfer yaitu antara lapisan E, lapisan sporadic, dan lapisan F.

### 2.1.2 *Skip Zone*

*Skip zone* merupakan istilah sering di dengar dalam sistem komunikasi radio HF. Istilah ini muncul karena adanya daerah yang tidak mendapatkan sinyal dari pemancar komunikasi HF yang di pantulkan melalui lapisan ionosfer, seperti ditunjukkan pada gambar 2.5. *Skip zone* terjadi ketika frekuensi yang digunakan dalam pegoprasian lebih besar daripada frekuensi kritis. Besaran *skip zone* dari suatu pemancar akan bergantung dari nilai frekuensi yang digunakan dan nilai frekuensi kritis pada lapisan ionosfer yang terjadi pemantulan<sup>[1]</sup>. Komunikasi yang terjadi pada daerah skip zone masih dapat diatasi menggunakan propagasi groundwave, namun jangkauannya terbatas pada jarak 50-100 km.

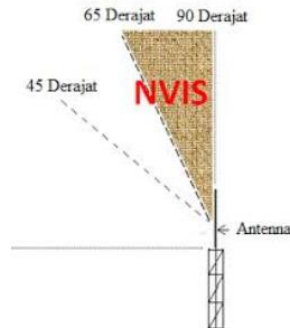


Gambar 2. 5 Daerah Skip Zone<sup>[11]</sup>

### 2.1.3 **Teknik *Near Vertical Incidence Skywave (NVIS)***

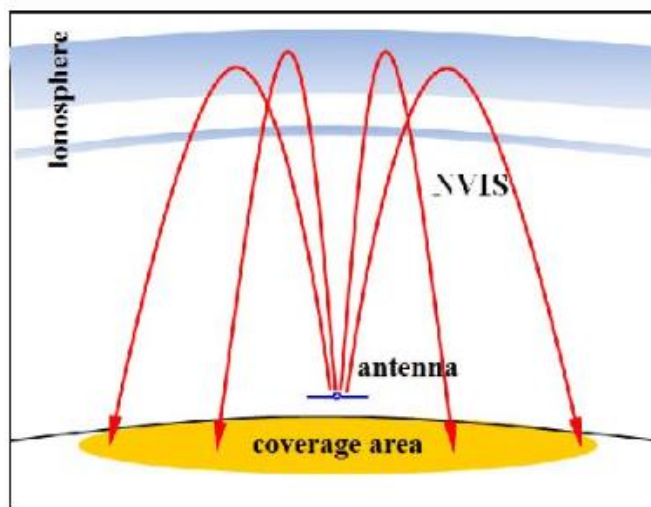
Near Vertical Incidence Skywave (NVIS) merupakan sistem komunikasi dengan teknik pemancaran gelombang radio HF hampir tegak lurus ke atas dengan sudut elevasi  $65^{\circ}$  sampai dengan  $90^{\circ}$ , seperti di tunjukkan pada gambar 2.6<sup>[12]</sup>. Dengan adanya sudut elevasi antenna yang hampir tegak lurus, maka teknik NVIS dapat diimplementasikan dalam sistem radio HF untk mengatasi dan menangani efek *skip zone*. Keberhasilan dalam mengimplementasikan sistem NVIS ini dipengaruhi

oleh 3 faktor, yaitu sudut elevasi, pemilihan frekuensi, dan besar keluaran daya pada sisi pemancar.



Gambar 2. 6 Sudut Elevasi Teknik NVIS <sup>[12]</sup>

Pada umumnya, penggunaan teknik NVIS pada lingkungan radio amatir ini biasanya menggunakan dua buah pita frekuensi, 7 MHz yang digunakan pada siang hari<sup>[4]</sup>. Propagasi sinyal radio HF dengan teknik NVIS dapat mencakup daerah hingga  $\pm 300$  km, seperti ditunjukkan pada gambar 2.7.



Gambar 2. 7 Propagasi Sinyal dengan Teknik NVIS <sup>[11]</sup>

Adapun beberapa kelebihan dari teknik NVIS adalah sebagai berikut :

- a. Daratan rendah atau lembah tidak menjadi masalah bagi propagasi dengan teknik NVIS.
- b. Tidak membutuhkan infrastruktur seperti repeater karena dua buah stasiun yang menggunakan teknik NVIS sudah dapat membangun sistem komunikasi yang handal tanpa dukungan dari pihak ketiga.

- c. Propagasi murni NVIS relatif bebas terhadap fading.
- d. Teknik NVIS dapat mengurangi noise dan interferensi, sehingga dapat meningkatkan besar *Signal to Noise Ratio* (SNR).

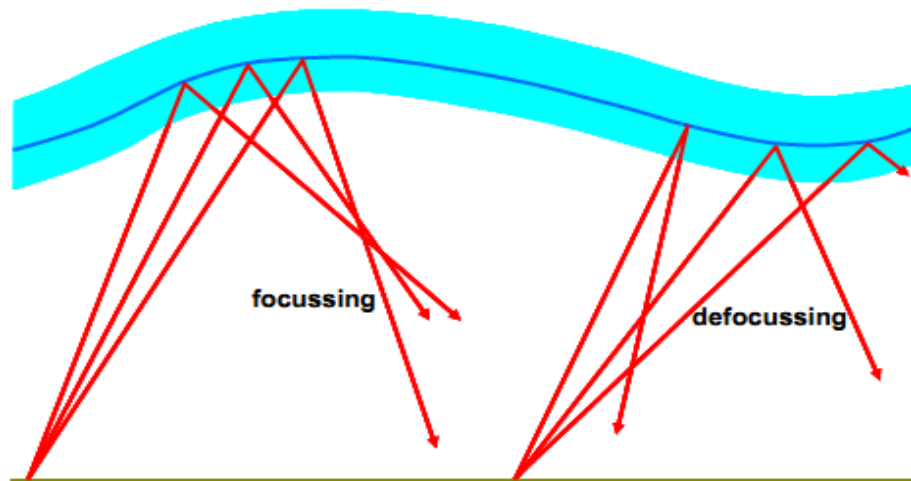
#### 2.1.4 Fading

Fading merupakan karakteristik utama dalam propagasi radio bergerak. Fading dapat didefinisikan sebagai perubahan fase, polarisasi dan level dari suatu sinyal terhadap waktu. Fading disebabkan oleh interferensi antara dua atau lebih versi dari sinyal yang diterima pada waktu yang sedikit berbeda sehingga level sinyal yang diterima berbeda-beda. Fading pada sistem komunikasi HF dapat disebabkan oleh multipath dimana terjadi penggabungan pada antena penerima untuk memberikan sinyal resultan yang dapat bervariasi dalam amplitudo dan fase, tergantung pada intensitas distribusi, waktu relatif propagasi dari gelombang dan bandwidth sinyal yang ditransmisikan.

Dalam komunikasi HF dikenal gangguan yang dikenal sebagai *Travelling Ionosphere Disturbance* (TID) mengakibatkan posisi sinyal bergeser sehingga sinyal menjadi fokus atau tidak fokus seperti pada gambar 2.8. *Travelling Ionosphere Disturbance* dapat menyebabkan variasi dalam fase, amplitudo, polarisasi dan sudut kedatangan gelombang radio<sup>[11]</sup>.

Fading bisa disebabkan oleh beberapa faktor penyebab diantaranya, yaitu<sup>[1]</sup> :

- a. Perubahan dari panjang lintasan propagasi dan pergerakan dari lapisan ionosfer.
- b. Perputaran dari polarisasi gelombang.
- c. Variasi absorpsi dari lapisan ionosfer.
- d. Menghilangnya sinyal dan ketidak fokusan lintasan sampai pada penerima disebabkan karena besarnya penurunan MUF dibawah frekuensi operasinya.



Gambar 2.8 Ketidakfokusan Lintasan Disebabkan Oleh Fading<sup>[11]</sup>

### 2.1.5 Frekuensi Kerja

Dalam ilmu Fisika, Pengertian frekuensi adalah jumlah jumlah getaran yang dihasilkan dalam setiap 1 detik. Sedangkan dalam ilmu elektronika, Frekuensi dapat diartikan sebagai jumlah gelombang listrik yang dihasilkan tiap detik. Frekuensi biasanya dilambangkan dengan huruf “f” dengan satannya adalah *Hertz* atau disingkat Hz. Jadi pada dasarnya 1 Hertz adalah sama dengan satu getaran atau satu gelombang listrik dalam satu detik. Istilah *hertz* ini diambil dari nama seorang fisikawan Jerman yaitu *Heinrich Rudolf Hertz* yang memiliki kontribusi dalam bidang elektromagnetisme. Salah satu hal terpenting dalam komunikasi HF adalah penentuan frekuensi yang digunakan. Penentuan frekuensi kerja pada sistem komunikasi HF salah satunya bisa dengan cara melakukan percobaan melakukan komunikasi antar dua radio untuk mendapatkan frekuensi yang bisa digunakan pada jarak yang sudah ditentukan, dan juga memperhitungkan frekuensi *Maximum Usable Frequency* (MUF) dan *Lowest Usable Frequency* (LUF). MUF merupakan frekuensi yang tercermin dari kerapatan elektron maksimum dalam lapisan ionosfer. MUF merupakan frekuensi tertinggi yang bisa dipantulkan oleh ionosfer dengan jarak tertentu antara dua buah stasiun radio yaitu pemancar dan penerima. Jika frekuensi lebih tinggi dari pada MUF maka frekuensi tersebut akan menembus lapisan ionosfer. Pada siang hari

dimungkinkan untuk berkomunikasi melalui lapisan E dan F menggunakan frekuensi yang berbeda. Frekuensi tertinggi yang didukung oleh lapisan E disebut EMUF, sedangkan frekuensi tertinggi yang didukung oleh lapisan F disebut FMUF. Besar nilai MUF bergantung pada frekuensi kritis dan sudut kedatangan antena yang akan digunakan. Dapat dikatakan semakin besar sudut *incidence* maka semakin besar nilai MUF semakin besar hingga mencapai tak terhingga. Dapat dilihat pada persamaan (2.1) digunakan untuk menghitung nilai MUF<sup>[1]</sup>.

Antena linear *array* memiliki persamaan *Array Factor* (AF) :

$$\text{MUF} = \frac{f_c}{\cos i} \quad (2.1)$$

Dimana :

$f_c$  = frekuensi critical

$i$  = angle of incidence

LUF (*Lowest Usable Frequency*) adalah frekuensi minimum yang digunakan agar sinyal sampai ke penerima. LUF terjadi ketika efisiensi antena menurun seiring menurunnya frekuensi. LUF juga dipengaruhi oleh lapisan ionosfer, khususnya oleh penyerapan sinyal suffer karena invers lapisan D. Penyerapan sinyal yang bergantung pada invers dari frekuensi, sehingga frekuensi akan turun, juga menyebabkan meningkatnya penyerapan sehingga daya sinyal di penerima menjadi lemah. Efek propagasi ini sangat tidak efisien, malah cenderung bersifat memperburuk sinyal di penerima. Karena LUF sangat tergantung secara signifikan dengan jumlah penyerapan sinyal suffer, maka nilai LUF menjadi sangat bervariasi. Dengan kata lain, nilai LUF tertinggi terdapat pada siang hari, di musim panas, dan pada saat matahari bersinar maksimum. Adapun LUF pada propagasi tergantung pada efisiensi antena, daya pancar transmitter dan berapa banyak penyerapan yang didapatkan oleh sinyal.

### 2.1.6 Polarisasi

Pada komunikasi HF terdapat dua macam antenna yang sering digunakan yaitu antenna vertical monopole dan antenna horizontal dipole. Pada antenna vertical monopole memiliki polarisasi vertikal dengan medan listrik yang tegak lurus dengan bidang ekuator, sedangkan pada antenna horizontal dipole memiliki polarisasi horizontal dengan medan listrik yang sejajar dengan bidang ekuator.

Pada saat gelombang radio dipancarkan dan dipantulkan oleh lapisan ionosfer maka gelombang tersebut terbagi menjadi dua buah gelombang karakteristik yang berpropagasi masing-masing di lapisan ionosfer. Kedua gelombang tersebut yaitu ordinary wave dan extraordinary waves yang berpolarisasinya secara elips atau elliptically polarized<sup>[13][14]</sup>. Dimana arah rotasi dari polarisasi medan ordinary dan extraordinary saling berlawanan.

### 2.1.7 Link Budget

Perhitungan *link budget* perlu dilakukan sebelum melakukan implementasi sistem pengukuran Radio HF, untuk dapat mengestimasi seberapa besar daya pancar yang dibutuhkan untuk suatu sistem komunikasi radio pada jarak tertentu. Pada *link* radio HF terdapat beberapa faktor yang mempengaruhi redaman pada lintasan *skywave*, hal tersebut ditunjukkan pada Tabel 2.1<sup>[1]</sup>.

Tabel 2.1 Jenis redaman diantara transmitter dan receiver antenna lintasan HF<sup>[1]</sup>.

Simbol	Nama Simbol	Nilai redaman (dB)
L <sub>a</sub>	Redaman <i>absorption</i>	0-20 per hop
L <sub>b</sub>	Redaman <i>free space</i>	5-120
L <sub>g</sub>	Redaman <i>ground reflection</i>	0-3 per pantulan
L <sub>p</sub>	Redaman <i>polarization coupling</i>	3-6
L <sub>q</sub>	Redaman <i>sporadic E</i>	0-9



Dimana:

- Redaman *absorption* ( $L_a$ )

Redaman *absorption* ( $L_a$ ) adalah redaman yang terjadi karena adanya penyerapan daya di lapisan E sehingga daya tidak dapat diteruskan ke lapisan F.  $L_a$  terdiri dari komponen deviatif dan non deviatif. Rugi-rugi ini dapat dihitung menggunakan persamaan 2.2 sebagai berikut :

$$L_a = \frac{1}{f^2} \quad (2.2)$$

Dimana:

$L_a$  = Redaman *absorption* (dB)

$f$  = frekuensi kerja (MHz)

- Redaman *free space* ( $L_b$ )

Pada saat sinyal meninggalkan antena, gelombang radio yang menyebar ke segala arah yang mengakibatkan pengurangan daya pada sisi penerima. Redaman propagasi *free space* atau yang sering disebut dengan FSL (*Free Space Loss*). Perhitungan redaman *free space* ini berguna untuk memprediksi besar daya sinyal di penerima. Berikut merupakan persamaan dari redaman *free space*.

$$L_b = 32.45 + 20\text{Log}10F(\text{MHz}) + 20\text{Log}10d(\text{km}) \quad (2.3)$$

Dimana:

$L_b$  = redaman *free space*(dB)

$d$  = jarak antara pemancar dan penerima (km)

$F$  = frekuensi (MHz)

- Redaman *ground reflection* ( $L_g$ )

Pada saat sinyal dipancarkan dari pemancar ke penerima dengan jarak yang jauh dimungkinkan terjadi redaman *ground reflection*. Redaman disebabkan oleh nilai konduktivitas dan dielektrik dari pantulan gelombang di permukaan bumi. Biasanya pantulan dipermukaan air laut

akan jauh lebih kecil dibandingkan dengan permukaan bumi yang kering. Redaman ini berkisar antara 0.5 sampai 3 dB.

- Redaman *polarization coupling* ( $L_p$ )

Redaman polarization coupling merupakan redaman yang disebabkan karena adanya perbedaan polarisasi antara gelombang yang diterima dengan antena di penerima. Dimana redaman ini berkisar antara 3 sampai 6 dB.

- Redaman *sporadic E* ( $L_q$ )

Redaman Sporadic-E disebabkan karena meningkatnya kerapatan elektron pada lapisan E. sehingga sinyal yang dipancarkan yang seharusnya dipantulkan oleh lapisan F, karena terdapat sporadic-E sinyal dipantulkan oleh oleh sporadic-E. Semakin tinggi nilai kerapatan elektron pada lapisan E, semakin tinggi probabilitas bahwa sinyal akan terhalang untuk mencapai lapisan F. Hal itu mengakibatkan jarak jangkauannya berubah sehingga sinyal yang dikirimkan tidak sampai pada penerima.

- Redaman Total ( $A_T$ )

Redaman total merupakan penjumlahan beberapa redaman pada lintasan komunikasi HF, besar redaman total pada lintasan komunikasi HF dapat ditulis sebagai berikut:

$$A_T = L_a + L_b + L_g + L_q \quad (2.4) \quad (2.1)$$

Dimana :

$A_T$  = Total Loss (dB)

$L_a$  = Redaman *absorption* (dB)

$L_b$  = Redaman *free space* (dB)

$L_g$  = Redaman *ground reflection* (dB)

$L_p$  = Redaman *polarization coupling* (dB)

$L_q$  = Redaman *sporadic E* (dB)

Perhitungan *Link budget* ini digunakan untuk dapat memperhitungkan besar daya pancar yang dibutuhkan, sehingga kualitas sinyal yang dikirimkan dapat diterima di penerima sesuai standar yang diinginkan.

$$P_{Rx} = P_{Tx} + G_{Tx} + A_T + G_{Rx} \quad (2.5) \quad (2.2)$$

Dimana :

$P_{Rx}$  = Daya pemancar (dBW)

$P_{Tx}$  = Daya penerima (dBW)

$G_{Tx}$  = Gain antena pemancar (dB)

$G_{Rx}$  = Gain antena penerima (dB)

$A_T$  = Redaman lintasan propagasi (*pathloss*) (dB)

## 2.2 Sistem Komunikasi Digital

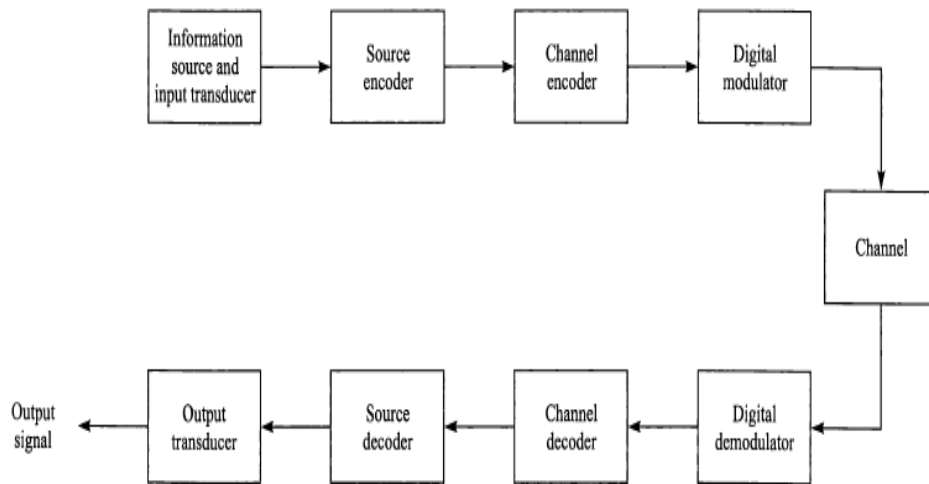
Sistem komunikasi digital semakin berkembang karena meningkatnya permintaan akan komunikasi data dan karena transmisi digital menawarkan pilihan pemrosesan data dan fleksibilitas yang tidak tersedia pada dengan transmisi analog<sup>[15]</sup>. Dalam sistem komunikasi digital, pesan yang dihasilkan oleh sumber diubah menjadi urutan angka biner (0 dan 1). Jika ingin merepresentasikan output (pesan) dengan sedikit mungkin angka biner, dengan kata lain, mencari representasi efisien dari output sumber yang menghasilkan redundansi sedikit atau tidak sama sekali. Proses konversi secara efisien keluaran dari sumber analog atau digital ke dalam urutan angka biner disebut source encoding atau data compression.

Urutan digit biner dari source encoder disebut information sequence, dilewatkan ke channel encoder. Tujuan dari ini adalah untuk memperkenalkan dalam urutan informasi biner yang dapat digunakan pada receiver untuk mengatasi efek kebisingan dan gangguan yang ditemui dalam transmisi sinyal melalui saluran. Urutan biner pada output di

channel encoder dilewatkan ke modulator digital yang berfungsi sebagai interface ke saluran komunikasi. Hampir semua saluran komunikasi yang dijumpai dalam praktiknya mampu mentransmisikan sinyal listrik (bentuk gelombang), dengan tujuan utama dari modulator digital adalah untuk memetakan urutan informasi biner ke dalam bentuk gelombang sinyal. Communication channel adalah media fisik yang digunakan untuk mengirim sinyal dari transmitter ke receiver. Dalam transmisi wireless, saluran tersebut mungkin merupakan atmosphere (free space).

Pada receiver sistem komunikasi digital, digital demodulator memproses gelombang yang ditransmisikan saluran yang rusak dan mengurangi bentuk gelombang menjadi urutan angka yang mewakili estimasi simbol data yang ditransmisikan (biner atau M-ary). Urutan angka ini diteruskan ke channel decoder yang mencoba merekonstruksi urutan informasi asli dari pengetahuan kode yang digunakan oleh channel decoder dan redundansi yang terkandung dalam data yang diterima.

Ukuran seberapa baik kinerja demodulator dan decoder adalah error frequency pada decode sequence. Lebih tepatnya, probability of bit-error pada output decoder adalah ukuran kinerja kombinasi demodulator-decoder. Secara umum, probability of error adalah fungsi dari karakteristik kode, jenis bentuk gelombang yang digunakan untuk mengirimkan informasi melalui saluran (jumlah noise&interference), daya transmisi, karakteristik saluran dan metode demodulasi dan decoding. Langkah terakhir ketika output analog yang diinginkan, source decoder menerima urutan output dari channel decoder dan dari pengetahuan tentang metode pengkodean sumber yang digunakan upaya untuk merekonstruksi sinyal asli dari sumber. Karena kesalahan decoding saluran dan kemungkinan distorsi yang diperkenalkan oleh source encoder dan mungkin, source decoder adalah prakiraan terhadap sinyal asli dan sinyal yang rekonstruksi adalah ukuran dari distorsi yang diperkenalkan oleh sistem komunikasi digital<sup>[16]</sup>. Seperti yang ditampilkan pada gambar 2.9 tentang komponen sistem komunikasi digital.



Gambar 2.9 Komponen sistem komunikasi digital<sup>[16]</sup>

### 2.2.1 *Low Pass Equivalent*<sup>[16]</sup>

Terdapat 2 cara untuk menyatakan sinyal *Low Pas Equivalent* (LPE) atau Complex Baseband. Pertama untuk sinyal Bandpass dengan modulasi amplitude (M-QAM) dapat dinyatakan pada persamaan 2.6. Kedua untuk sinyal Bandpass dengan modulasi fase (M-PSK), LPE dapat dinyatakan dalam bentuk persamaan 2.7.

$$r_e = x_I + jx_Q \quad (2.6)$$

Dimana :

$x_I$  sebagai *in-phase*

$x_Q$  sebagai *quadrature*

$$r_e = \rho e^{j\theta} \quad (2.7)$$

Dimana :

$\rho$  sebagai magnitudo

$\theta$  sebagai *phase*

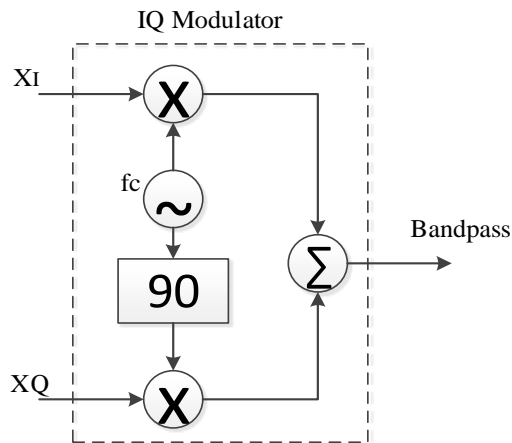
Dengan demikian misalkan  $r(t)$  adalah sinyal bandpass (*modulated carrier*) maka hubungan antara  $r(t)$  dengan sinyal *lowpass equivalent* atau *complex baseband*  $r_e(t)$  dapat dituliskan pada persamaan 2.8 sebagai berikut.

$$r(t) = \text{Re} \{r_e(t)e^{j\omega_c t}\}, \omega_c = 2\pi f_c \quad (2.8)$$

Dengan demikian maka bentuk dari sinyal *bandpass* dengan *Low Pass Equivalent*  $r_e(t)$  dapat di tuliskan pada persamaan 2.9 dan 2.10. Selain itu juga dapat digambarkan blok diagram *Low Pass Equivalent* pada gambar 2.10.

$$\begin{aligned}
 r(t) &= \text{Re} \{r_e(t)e^{j\omega t}\} \\
 &= \text{Re} \{(x_I + jx_Q)e^{j\omega t}\} \\
 &= \text{Re} \{(x_I + jx_Q)(\cos\omega t + j\sin\omega t)\} \\
 &= \text{Re} \{(x_I\cos\omega t + jx_I\sin\omega t + jx_Q\cos\omega t - x_Q\sin\omega t)\} \\
 &= \text{Re} \{(x_I\cos\omega t - x_Q\sin\omega t) + j(x_I\sin\omega t + jx_Q\cos\omega t)\} \\
 &= x_I\cos\omega t - x_Q\sin\omega t \\
 &= x_I \cos 2\pi fct - x_Q \sin 2\pi fct \tag{2.9}
 \end{aligned}$$

$$\begin{aligned}
 &= \text{Re} \{\rho e^{j\theta} e^{j\omega t}\} \\
 &= \text{Re} \{\rho e^{j(\omega t + \theta)}\} \\
 &= r(t)\text{Re} \{\rho((\cos(\omega t + \theta) + j\sin(\omega t + \theta)))\} \\
 &= \text{Re} \{\rho\cos(\omega t + \theta) + j\rho\sin(\omega t + \theta)\} \\
 &= \rho\cos(\omega t + \theta) \\
 &= 1\cos(2\pi fct + \frac{\pi}{4}) \\
 &= \frac{1}{\sqrt{2}}\cos(2\pi fct) - \frac{1}{\sqrt{2}}\sin(2\pi fct) \approx 1\cos(2\pi fct + \frac{\pi}{4}) \tag{2.10}
 \end{aligned}$$

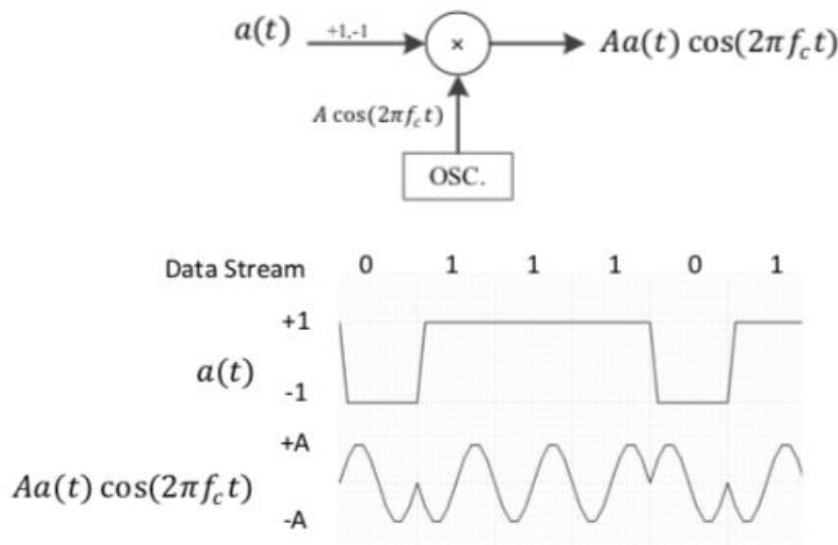


Gambar 2.10 Blok Diagram *Low Pass Equivalent*

### 2.2.2 Modulasi Binary Phase-Shift Keying (BPSK)

*Phase-Shift Keying* (PSK) merupakan salah satu teknik modulasi digital yang paling efisien dan secara luas digunakan di dalam sistem komunikasi modern seperti *satellite link* dan *wideband microwave radio relay systems*. Contoh penggunaan PSK berdasarkan penggunaan jumlah pergeseran fasanya adalah BPSK dan *Quadrature Phase-Shift Keying* (QPSK).

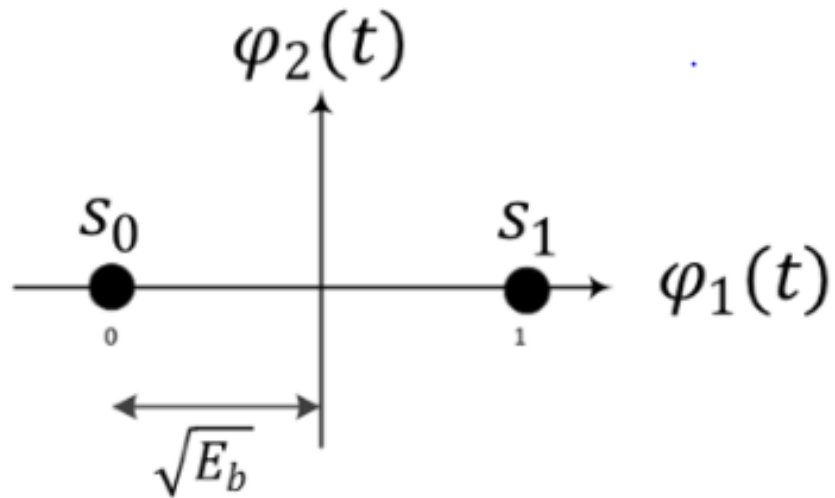
Pada sistem modulasi BPSK, sinyal termodulasi merupakan sinyal sinusiodal dengan frekuensi dan amplitudo tetap dalam fasanya, seperti di tunjukkan pada gambar 2.11. Jika informasi adalah 0 bit, maka sinyal modulasi akan tetap dalam fasanya, begitu juga sebaliknya. Apabila informasi adalah 1 bit, maka sinyal modulasi akan membalik fasanya sebesar  $180^\circ$ <sup>[17]</sup>.



Gambar 2.11 BPSK Modulator dan Waveform<sup>[17]</sup>

BPSK memiliki dua sinyal yang dipilih  $S_1(t)$  dan  $S_2(t)$  dalam persamaan 2.11 merepresentasikan 1 simbol digital. Kedua sinyal ini berada dalam frekuensi yang sama dan membawa jumlah energi yang sama walaupun memiliki koefisien korelasi sebesar -1.

$$\begin{aligned}
 S_1(t) &= A \cos(2\pi f_c t), \text{ simbol} = 1 \\
 S_2(t) &= -A \cos(2\pi f_c t), \text{ simbol} = 0
 \end{aligned}
 \tag{2.11}$$



Gambar 2.12 BPSK Constellation<sup>[17]</sup>

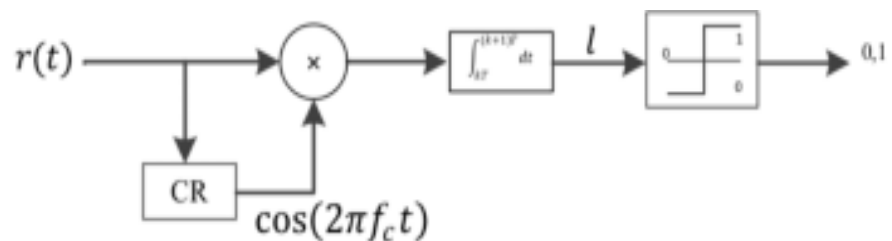
Pada gambar 2.12 menunjukkan kombinasi linear dari fungsi basis orthonormal  $\varphi_1(t)$  dan  $\varphi_2(t)$  seperti pada persamaan 2.12. Dalam persamaan ini,  $S_1(t)$  dan  $S_2(t)$  mewakili sinyal yang memiliki energy of bit  $E_b$  yang diberikan pada persamaan 2.13<sup>[17]</sup>.

$$\varphi_1(t) = \sqrt{\frac{2}{T_b}} \cos(2\pi f_c t)$$

$$\varphi_2(t) = -\sqrt{\frac{2}{T_b}} \cos(2\pi f_c t) \quad (2.12)$$

$$E_b = \frac{A^2 T_b}{2} \quad (2.13)$$

Untuk menghasilkan sistem demodulator BPSK yang koheren, sinyal referensi harus disinkronkan dengan frekuensi dan fasa dari sinyal yang diterima  $r(t)$ . Dalam penerapannya, *carrier recovery block* bekerja menghasilkan sinyal referensi seperti pada gambar 2.13 sebagai berikut.



Gambar 2.13 BPSK Demodulator<sup>[17]</sup>



### 2.2.3 *Signal to Noise Ratio (SNR)*

Berdasarkan Teori Nyquist, agar tidak terjadi kesalahan pada deteksi sinyal, filter harus memiliki bandwidth tidak kurang dari setengah laju symbol. Adapun karena penerima terjadi kesalahan dapat dinyatakan dengan probabilitas error. Probabilitas error BPSK dinyatakan dengan persamaan 2.14 sebagai berikut.

$$Pe = Q\left(\sqrt{\frac{2Eb}{No}}\right) \quad (2.14)$$

Dimana nilai Eb/No dapat diketahui menggunakan persamaan 2.15 sebagai berikut.

$$\frac{Eb}{No} = \frac{S}{N} \frac{W}{R} \quad (2.15)$$

Adapun S merupakan daya sinyal, N merupakan daya noise, W merupakan bandwidth dalam satuan hertz dan R merupakan laju bit dalam satuan bit/detik. Penentuan SNR bisa didapat dengan persamaan 2.16 sebagai berikut<sup>[16]</sup>.

$$SNR(\text{dB}) = 10 \log_{10} \frac{S}{N} \quad (2.16)$$

### 2.2.4 **Indikator Kinerja**

Pada indicator kinerja terdapat beberapa indikator diantaranya *Bit Error Rate (BER)*, *Character Error Rate (CER)* dan *Frame Error Rate (FER)* sebagai berikut.

#### 2.2.4.1 *Bit Error Rate (BER)*

*Bit Error Rate* adalah suatu rumusan kesalahan laju bit yang terjadi selama pentransmisian data antara terminal pengirim dan terminal penerima. Kesalahan yang terjadi selama proses pentransmisian data antara masukan dan keluaran dilakukan dengan cara membandingkan masukan total yang dikirim dengan jumlah bit yang error. Persamaan *Bit Error Rate* dapat dihitung pada persamaan 2.17 sebagai berikut .

$$BER = \frac{\text{Jumlah Bit Error}}{\text{Jumlah Bit Yang Dikirim}} \quad (2.17)$$

#### 2.2.4.2 Character Error Rate (CER)

CER merupakan singkatan *Character Error Rate*. Karakter adalah setiap huruf, angka, tanda baca seperti titik, koma tanda seru, tanda tanya, tanda kutip dan juga spasi. CER merupakan parameter yang dipakai untuk mengukur tingkat akurasi data dari pesan yang dikirimkan dengan cara menghitung jumlah karakter yang rusak dibandingkan dengan karakter keseluruhan. Persamaan *Character Error Rate* dapat dihitung pada persamaan 2.18 sebagai berikut.

$$CER = \frac{\text{Jumlah Karakter Error}}{\text{Jumlah Karakter Yang Dikirim}} \quad (2.18)$$

#### 2.2.4.3 Frame Error Rate (FER)

Frame berupa paket yang terdiri dari beberapa karakter. Satu Frame dalam Protokol AX.25 terdiri dari 255 bit<sup>[18]</sup>. FER dapat dihitung menggunakan persamaan 2.19 sebagai berikut.

$$FER = \frac{\text{Jumlah Frame Error}}{\text{Jumlah Frame Yang Dikirim}} \quad (2.19)$$

#### 2.2.5 Filter Raised Cosine

Salah satu *filter* yang banyak digunakan untuk menghasilkan tanggapan impuls dengan titik-titik perpotongan nol yang berjarak sama, dan tanggapan frekuensi yang kedua sisi kiri dan kananya berangsur-angsur turun menjadi nol, ketimbang patah dan jatuh seketika nol, adalah *filter raised-cosine*. Tanggapan frekuensi sebuah filter *raised-cosine* dapat dituliskan sebagai berikut pada persamaan 2.20.

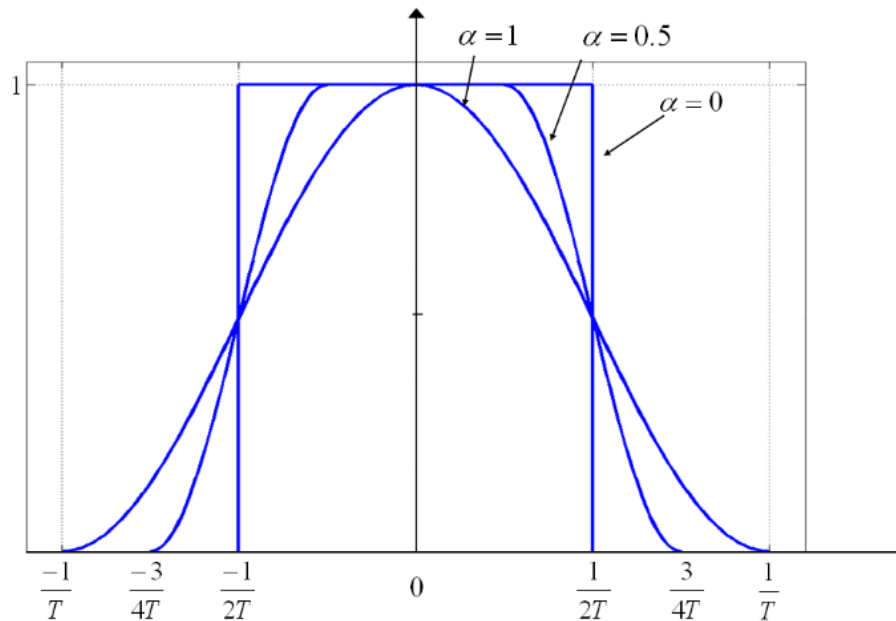
$$H(\omega) = \begin{cases} \frac{1}{2} \left\{ 1 - \sin \left[ \frac{\pi}{2\alpha W} (|\omega| - W) \right] \right\} & 0 \leq |\omega| \leq (1 - \alpha)W \\ \frac{1}{2} \left\{ 1 + \sin \left[ \frac{\pi}{2\alpha W} (|\omega| - W) \right] \right\} & (1 - \alpha)W \leq |\omega| \leq (1 + \alpha)W \\ 0 & |\omega| > (1 + \alpha)W \end{cases} \quad (2.20)$$

Di diman  $W = \pi T_s$ . Dengan nilai impulsnya adalah sebagai berikut pada persamaan 2.21.

$$h(t) = \frac{1}{T_s} \left( \frac{\sin Wt}{Wt} \right) \left[ \frac{\cos \alpha Wt}{1 - (2\alpha Wt/\pi)^2} \right] \quad (2.21)$$

Pada persamaan 2.21 memiliki bentuk umum  $(\sin x)/x$ , yang biasanya digunakan pada impuls *filter low-pass* ideal.

Pada gambar 2.14 plot  $H(\omega)$  dan  $h(t)$  untuk tiga nilai  $\alpha$  yang berbeda. Parameter  $\alpha$  disebut sebagai faktor hamparan (*roll-of factor*). Pada kasus  $\alpha = 0$  adalah karakteristik dari filter low-pass ideal. Pada kasus  $\alpha = 1$  dikenal sebagai karakteristik kosinus<sup>[19]</sup>.



Gambar 2.14 *Filter Raised-Cosine Dengan Ragam Roll-Of Factor*<sup>[19]</sup>

Berdasarkan gambar 2.13 dapat diketahui *bandwidth* yang diduduki oleh sebuah sinyal digital yang memiliki karakteristik kosinus, minimum dari  $f_b=1/2T$  untuk  $\alpha = 0$  hingga maksimum  $f_b=1/T$  untuk  $\alpha = 1$ . Dengan demikian kebutuhan *bandwidth* dapat dicari menggunakan persamaan 2.22 sebagai berikut<sup>[19]</sup>.

$$f_b = \frac{1+\alpha}{2T} \text{ Hz} \quad (2.22)$$

### 2.2.6 *Low Pass Filter*

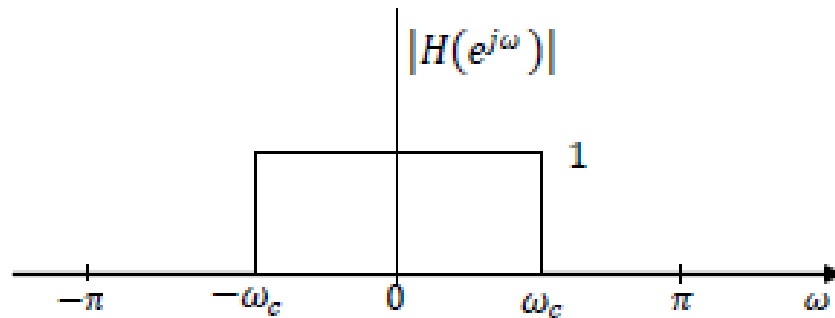
Filter digital sering disebut sebagai sistem diskrit. Dimana filter dapat diklasifikasikan berdasarkan respons frekuensinya yaitu filter fasa linier, filter *allpass*, dan filter selektif frekuensi. Pada filter selektif frekuensi terdapat beberapa jenis filter diantaranya yaitu LPF (*Low Pass Filter*), HPF (*High Pass Filter*), BPF (*Band Pass Filter*) dan BSF (*Band Stop Filter*). Interval frekuensi pada respons magnituda yang bernilai 1 atau konstan disebut daerah *passband* (pita lolos) sedangkan interval

frekuensi pada respons magnituda yang bernilai 0 disebut daerah stopband. Frekuensi yang membatasi *passband* dan *stopband* disebut frekuensi *cutoff*.

LPF mempunyai respons magnituda seperti pada Gambar 2.15 dan selalu periodik dengan periode  $2\pi$ . LPF mempunyai frekuensi *cutoff* dan secara matematik dapat ditulis sebagai berikut:

$$H(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \omega_c < |\omega| \leq \pi \end{cases}$$

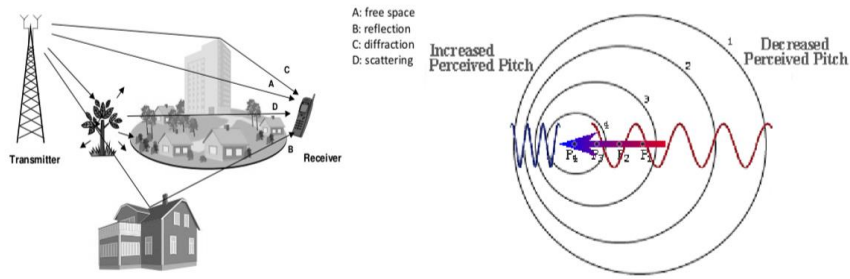
$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega = \frac{\sin \omega_c n}{\pi n} \quad (2.23)$$



Gambar 2.15 Filter LPF Ideal

### 2.2.7 Sinkronisasi Gelombang Pembawa

Selama proses transmisi, sinyal melintas melalui jalur yang berbeda. Peristiwa ini terjadi karena mekanisme propagasi gelombang radio, seperti refleksi dan difraksi yang disebut sebagai *multipath effect*<sup>[17]</sup>. Hal ini mengakibatkan adanya perubahan fasa yang dapat menyebabkan perubahan pendeteksian pada sisi penerima. Selain *multipath effect*, fenomena lain juga dapat menyebabkan pergeseran pendeteksian frekuensi pada sisi penerima, fenomena ini disebut dengan *Doppler frequency shift*<sup>[20]</sup>. Fenomena ini terjadi karena sistem komunikasi HF memanfaatkan lapisan ionosfer untuk propagasinya, di mana elektron pada lapisan ionosfer selalu berubah tiap waktunya<sup>[1]</sup>. Perubahan frekuensi dan fasa ini dapat dikompensasi dengan adanya sinkronisasi gelombang pembawa.

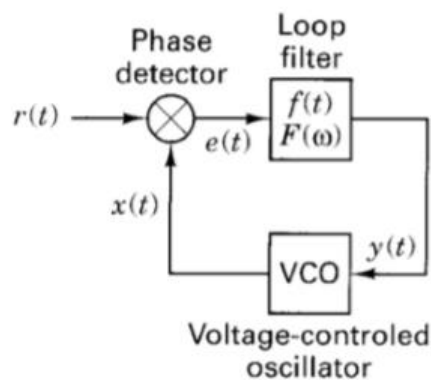


Gambar 2.16 *Multipath Effect* dan *Doppler Shift*

Dalam sistem komunikasi, terkhusus untuk penerima coherent, sinkronisasi sangat dibutuhkan. Sinkronisasi terdiri menjadi tiga tingkatan, sinkronisasi gelombang pembawa, sinkronisasi simbol, dan sinkronisasi frame<sup>[15]</sup>. Di mana salah satu masalah umum dalam komunikasi tanpa kabel adalah sinkronisasi gelombang pembawa dengan menggunakan PLL yang merupakan inti dari sinkronisasi gelombang pembawa.

### 2.2.7.1 *Phase-Locked Loop (PLL)*

PLL merupakan sistem feedback loop yang tersusun atas *Phase Detector (PD)*, *Loop Filter (LF)* yang merupakan *Low Pass Filter (LPF)*, dan *Voltage Controlled Oscillator (VCO)*<sup>[19]</sup>. VCO dan Phase Detector saling terhubung, sehingga oscillator dapat menjaga sudut fasa konstan relatif terhadap sinyal referensi seperti pada gambar 2.17. Cakupan dari rangkaian ini adalah untuk menjaga koherensi antara sinyal pada frekuensi  $r(t)$  dengan sinyal yang dikeluarkan oleh VCO,  $x(t)$ , untuk dibandingkan perubahan frekuensi dan fasanya.



Gambar 2.17 Skema Dasar PLL<sup>[21]</sup>

### 2.2.7.2 Phase Detector(PD)

Fungsi dari PD adalah untuk membandingkan fasa dari sinyal pembawa yang masuk, yaitu terdapat pada persamaan 2.24 sebagai berikut.

$$r(t) = \cos(2\pi f_1 t + \alpha) \quad (2.24)$$

Dengan sinyal replika keluaran dari VCO, yaitu terdapat pada persamaan 2.25.

$$x(t) = -\sin(2\pi f_2 t + \hat{\alpha}) \quad (2.25)$$

Maka, *output* yang dihasilkan blok PD adalah  $e(t)$  terdapat pada persamaan 2.26 sebagai berikut.

$$\begin{aligned} e(t) &= [\cos(2\pi f_1 t + \alpha) X [-\sin(2\pi f_2 t + \hat{\alpha})] \\ e(t) &= -\frac{1}{2} \sin[(2\pi(f_1 + f_2))t + \alpha + \hat{\alpha}] + \\ &\quad \frac{1}{2} \sin [(2\pi(f_1 - f_2))t + \alpha - \hat{\alpha}] \end{aligned} \quad (2.26)$$

### 2.2.7.3 Loop Filter (LF)

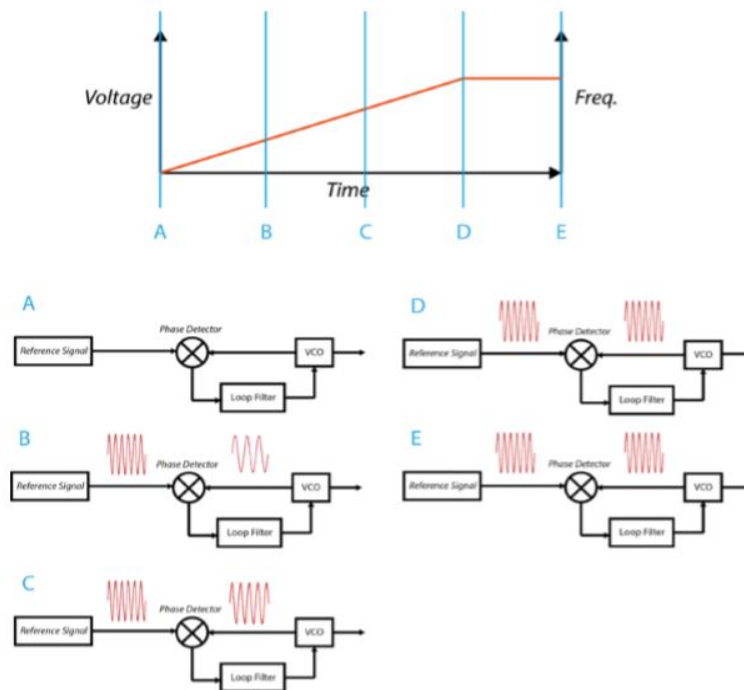
Karena hasil sinyal dari blok PD,  $e(t)$ , selalu berubah-ubah, sehingga mengakibatkan perbedaan fasa menjadi time-varying signal ke dalam blok *Loop Filter* (LF). Blok ini umumnya berisi *Low Pass Filter* (LPF) yang berfungsi untuk menyaring output dari PD dan menentukan karakteristik untuk VCO. Jika PD menghasilkan output, maka LF akan mengintegrasikan sinyal output dari PD ke tegangan DC untuk menggerakkan VCO ke frekuensi tertentu <sup>[22]</sup>.

$$y(t) = e(t) * f(t) \quad (2.27)$$

### 2.2.7.4 Voltage-Controlled Oscillator (VCO)

VCO merupakan sebuah oscillator yang hasil keluaran frekuensinya berasal dari input tegangan. Tegangan yang dihasilkan dari blok LF akan menyesuaikan dengan frekuensi dan fasa replika dari VCO dan frekuensi dan fasa sinyal yang diterima. Maka, sinyal keluaran dari VCO<sup>[20]</sup> terdapat pada persamaan 2.28.

$$x(t) = \cos[w_0 t + \int_0^t y(t)(\phi)d\phi] \quad (2.28)$$



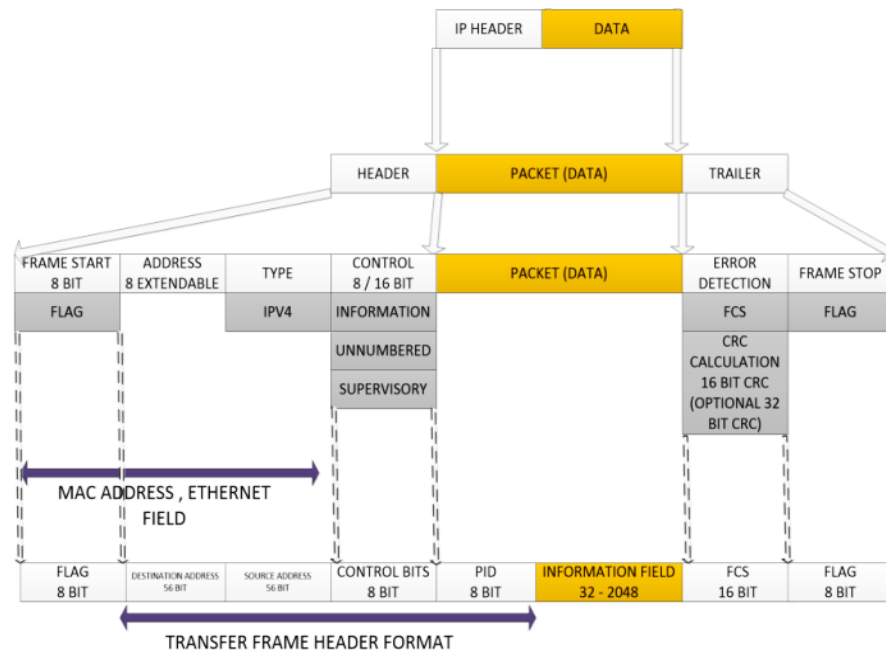
Gambar 2.18 Proses Kondisi *Steady State* Sistem PLL

Apabila blok LF sudah mendeteksi minimnya phase error, maka loop akan terkunci. Sehingga yang terjadi adalah situasi steady state yang mana PLL akan mengeluarkan sinyal dengan input yg sama selama kondisi ini, seperti pada Gambar 2.18.

### 2.2.8 Protocol AX.25

AX.25 (Amateur AX.25) merupakan protokol layer 2 yaitu terdapat pada Data Link Layer. Sebagai protokol layer 2 AX.25 bertanggungjawab untuk membangun *link connection*, dimana menyediakan prosedur *logic* untuk *information transfer*, dan *link disconnection*. Protokol AX.25 sedikit berbeda dengan protokol lain, karena pada protokol ini menghilangkan perbedaan kelas dari perangkat yang terhubung pada *link*, sedangkan beberapa protokol pada *layer datalink* membedakan kelas dari perangkat yang terhubung. Sebagai protokol yang bertanggung jawab dalam membangun dan memutuskan

link dan mengirim informasi, protokol ini dapat digunakan dalam sistem yang bekerja pada frekuensi tinggi yang terdapat *noise*.



Gambar 2.19 Ilustrasi Struktur Protokol Ax.25

Protokol ini menggunakan prinsip – prinsip yang terdapat dalam International Standards Organization (ISO) Information Standards (IS) 3309, 4335 dan 7809 tentang HDLC. Dapat diilustrasikan pada gambar 2.19<sup>[18]</sup>.

Protokol AX25 merupakan protokol untuk melakukan akses jaringan sinkron antara DTE (*Data Terminal Equipment*) pada sisi pemakai dan DCE (*Data Circuit Terminating Equipment*) yang merupakan peralatan yang berada pada sisi jaringan yang langsung berhubungan dengan sisi pemakai. Link level, merupakan layer yang mempunyai aturan untuk bertukar data/*datalink control*. Protokol yang dipakai pada lapisan ini adalah ISO disebut HDLC (*High Level Datalink Control*) yang melakukan beberapa hal sebagai berikut.

1. Membangun hubungan logik melalui media yang ada seperti kabel atau atmosfer.
2. Memberikan informasi mengenai perpindahan data agar data tetap pada urutannya.



3. Melakukan pendeteksian kesalahan.
4. Menutup hubungan logik yang telah selesai digunakan<sup>[18]</sup>.

### 2.2.8.1 Struktur Frame AX.25

Protokol AX.25 mengirimkan data dalam bentuk frame dan tiap frame tersusun dari bagian- bagian yang lebih kecil (*field*). Setiap *field* memiliki fungsi khusus dan terdiri dari jumlah byte. Dalam protokol AX.25 terdapat 3 tipe frame :

1. Frame Informasi  
Fungsi frame ini adalah untuk membawa data informasi yang dikirim oleh pengirim dan diterima oleh penerima.
2. Frame Pengawas atau Supervisory frame (Frame S)  
Fungsi frame ini adalah untuk melakukan pengendalian link.
3. Frame tidak bernomor atau Unnumbered Frame (Frame U)

Fungsi frame ini adalah untuk melakukan pengendalian link yang belum dilakukan oleh frame pengawas seperti membangun hubungan.

Tabel 2.2 Bentuk Frame U dan S

Flag	Address	Control	Info	FCS	Flag
8 bits	112/224 bits	8/16 bits	N*8 bits	16 bits	8 bits

Tabel 2.3 Bentuk Frame Informasi

Flag	Address	Control	PID	Info	FCS	Flag
8 bits	112/224 bits	8 / 16 bits	8 bits	N*8 bits	16 bits	8 bits

Fungsi dari masing – masing field adalah sebagai berikut :

1. Field Flag  
Field flag memiliki fungsi untuk menentukan awal dan akhir dari suatu frame. Hal ini memudahkan penerima untuk mengidentifikasi suatu frame. Field ini berisi satu octet yang bernilai 01111110 (7E hexa).

Urutan bit field flag tidak diperbolehkan muncul pada bagian manapun dalam frame yang telah lengkap. Untuk itu digunakan fasilitas bit stuffing yang setiap kali terdapat lima bit “1” muncul maka bit selanjutna diset 0.

## 2. Field Address

Fungsi dari field ini adalah sebagai pengalamatan dalam komunikasi. Field ini dapat mengidentifikasi stasiun pengirim, dan pemancar. Masing-masing subfield alamat pengirim dan penerima terdiri dari tujuh octet, enam diawal merupakan callsign yang merupakan susunan karakter abjad dan karakter American Standard Code for Information Interchange (ASCII), dan octet ketujuh merupakan Secondary Station Identifier (SSID) yang menyatakan nomor terminal dalam suatu stasiun.

Tabel 2.4 Address Subfield

<i>Address Subfield</i>						
A	A	A	A	A	A	A
1	2	3	4	5	6	7
Call Sign						SSID

Pada Tabel diatas A1 sampai dengan A6 merupakan octet yang berisi *callsign*, dan A7 merupakan octet yang berisi SSID. Contohnya adalah tabel 2.4 menunjukkan octet A1 merupakan octet yang dikirim terlebih dahulu.

Tabel 2.5 Address Field

<b>Octet</b>	<b>ASCII</b>	<b>Bin Data</b>	<b>Hex Data</b>
A1	N	10011100	98
A2	J	10010100	94
A3	7	01101110	6E
A4	P	10100000	A0
A5	Space	01000000	40
A6	Space	01000000	40
A7	SSID	11100000	E0

Pada *Callsign octet* terakhir diisi dengan spasi karena jumlah *callsign* kurang dari 6 karakter. Bit “R” merupakan bit cadangan, menjadi 1 apabila tidak digunakan. Sedangkan bit “C” merupakan bit perintah dari frame. Dalam field address terdapat 2 mode yaitu “*Non-repeater address field encoding*” dan mode “*repeater address field encoding*”. Mode *non-repeater address field encoding*

Mode ini digunakan saat *repeater* tidak digunakan. Pada mode ini field alamat terdiri dari 14 oktet yang kemudian dibagi menjadi 2 subfield alamat yaitu subfield alamat tujuan dan subfield alamat sumber.

Tabel 2.6 Address Field of Frame on

<b><i>Address Field of Frame</i></b>													
Subfield Alamat Tujuan							Subfield Alamat Sumber						
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14

*Non repeater address field*

Alamat tujuan terdiri dari oktet A1 sampai dengan A7. Oktet A1 sampai A6 merupakan *callsign* amatir dan oktet A7 merupakan SSIS dari stasiun radio amatir yang dikirimkan frame. Sedangkan untuk alamat sumber terdiri dari A8 sampai dengan A14. Dimana A8 sampai A13 merupakan *callsign* amatir dan oktet A14 merupakan SSID dari stasiun yang mengirim frame. Pada tabel..merupakan contoh frame AX.25 dengan mode non-repeater menggunakan frame Information (I).

Tabel 2.7 Frame AX.25 mode non-repeater

Octet	ASCII	Bin Data	Hex Data
Flag		01111110	7E
A1	N	10011100	98
A2	J	10010100	94
A3	7	01101110	6E
A4	P	10100000	A0
A5	Space	01000000	40
A6	Space	01000000	40
A7	SSID	11100000	E0

Tabel 2.7 Frame AX.25 mode non-repeater (lanjutan)

Octet	ASCII	Bin Data	Hex Data
A8	N	10011100	98
A9	7	01101110	6E
A10	L	10011000	98
A11	E	10001010	8A
A12	M	10011010	9A
A13	Space	01000000	40
A14	SSID	01100001	61
Control	1	00111110	3E
PID	None	11110000	F0
FCS	Part1	XXXXXXXXXX	HH
FCS	Part2	XXXXXXXXXX	HH
Flag	Part2	01111110	7E

### 2.2.8.2 Mode Repeater Address Field Encoding

Mode ini digunakan pada pengiriman frame melalui repeater. Pada mode ini alamat repeater ditambahkan dalam address field sehingga bertambah menjadi 21 oktet yang terbagi menjadi 3 subfield.

Tabel 2.8 Mode Repeater Address Field

<i>Address Field of Frame</i>								
Subfield Alamat Tujuan			Subfield Alamat Sumber			Subfield Alamat Repeater		
A1	A2	A3	A8	A9	A10	A15	A16	A17 A18
A4	A5	A6	A11	A12	A13	A19	A20	A21
A7			A14					

Alamat tujuan dikodekan dengan A1 hingga A7 dimana A1 sampai A6 adalah callsign amatir dan A7 adalah SSID stasiun. Alamat sumber dikodekan dengan A8 hingga 14 dimana A8 hingga A13 merupakan callsign amatir dan A14 merupakan SSID stasiun, alamat repeater dikodekan dengan A15 sampai dengan A21, dimana A15 sampai A20 merupakan callsign amatir dan A21 merupakan SSID stasiun. Protokol AX25 dapat menggunakan 2 repeater dengan panjang field address 28 oktet.

Tabel 2.9 Frame Protokol AX.25 Dengan Satu Alamat Repeater

Octet	ASCII	Bin Data	Hex Data
Flag		01111110	7E
A1	N	10011100	98
A2	J	10010100	94
A3	7	01101110	6E
A4	P	10100000	A0
A5	Space	01000000	40
A6	Space	01000000	40
A7	SSID	11100000	E0
A8	N	10011100	98
A9	7	01101110	6E
A10	L	10011000	98
A11	E	10001010	8A
A12	M	10011010	9A
A13	Space	01000000	40
A14	SSID	01100001	61
A15	N	10011100	98
A16	7	01101110	6E
A17	0	10011110	9E
A18	0	10011110	9E
A19	Space	01000000	40
A20	Space	01000000	40
A21	SSID	11100011	E3
Control	I	00111110	3E
PID	None	11110000	F0
FCS	Part1	XXXXXXXX	HH
FCS	Part2	XXXXXXXX	HH
Flag	Part2	01111110	7E

### 2.2.8.3 Control Field

Field ini berfungsi untuk mengidentifikasi tipe frame yang digunakan pada koneksi layer 2. Field terdiri dari 8 atau 16 bit. Dimana pada AX.25 terdapat tiga format control field:

a. *Information Field (I Frame)*

Semua frame Informasi memiliki bit 0 dari control field diatur ke "0".

Pada frame ini terdapat N(S) dan N(R). Dimana N(S) adalah nomor

urut pengiriman dari pengirim. Dan N(R) merupakan nomor penerimaan.

b. *Supervisory Field (S Frame)*

Frame supervisory memiliki bit 0 dari control field diatur ke “1” dan bit 1 dari control field diatur ke “0”. Frame S menyediakan Supervisory Link Control. Pada frame ini juga terdapat N(S) dan N(R).

c. *Unnumbered Field (U Frame)*

Unnumbered Frame kedua bit 0 dan 1 dari control field diatur ke “1”. Frame U bertanggung jawab untuk menjaga control tambahan atas link melampaui apa yang dicapai dengan frame S. Frame U bertanggung jawab untuk membangun dan mengakhiri koneksi link.

Tabel 2.10 Bentuk *Control Field* (Modulo 8)

Control Field Type	Control Field Bits							
	7	6	5	4	3	2	1	0
I Frame	N(R)			P	N(S)			0
S Frame	N(R)			P/F	S	S	0	1
U Frame	M	M	M	P/F	M	M	1	1

Tabel 2.11 Bentuk *Control Field* (Modulo 128)

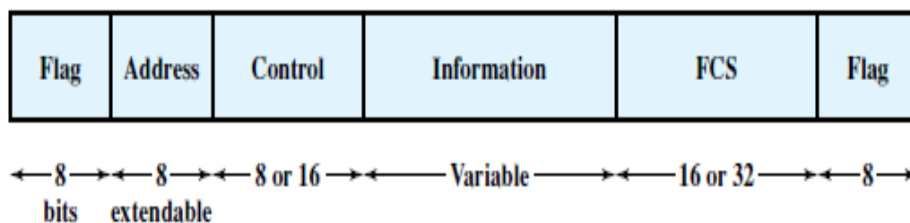
Control Field Type	Control Field Bits													
	15	14	13	12	11	10	9	4	3	2	1	0		
I Frame	N(R)						P	N(S)			0			
S Frame	N(R)						P/F	0	0	0	S	S	0	1

Tabel 2.10 dan 2.11 merupakan dasar dari control field yang terkait dengan masing masing tiga jenis frame. Control field dapat menjadi satu atau dua octet panjang dan dapat menggunakan nomor urut untuk menjaga

keutuhan link. Nomor urut ini dapat berupa tiga-bit (modulo 8) atau tujuh-bit (modulo 128). Pertama, bit yang dikirim adalah bit 0, sedangkan untuk bit yang terakhir dikirim adalah bit ke 7 untuk modulo 8 dan bit ke 15 untuk modulo 128. N(S) merupakan nomor urut pengiriman dan N(R) merupakan nomor urut penerimaan. Bit “S” menyatakan frame supervisory dan bit “M” menyatakan frame unnumbered, dan bit P/F (Poll/Final) digunakan pada semua jenis frame.

### 2.2.9 Sinkronisasi Data Pada *Protocol AX.25*

Sinkronisasi pada protokol AX.25 menggunakan struktur frame HDLC. HDLC menggunakan synchronous transmission. Semua transmisi dalam bentuk frame, dan format frame tunggal sudah cukup untuk semua jenis pertukaran data dan kontrol. Pada gambar 2.20 menjelaskan struktur frame HDLC. *Flag*, *address*, dan *control field* yang mendahului bidang informasi dikenal sebagai *header*. FCS dan flag yang mengikuti bidang data disebut sebagai *trailer*<sup>[23]</sup>.



Gambar 2.20 Format Frame

*Flag field* membatasi *frame* di kedua ujungnya dengan pola unik 01111110. *Singleflag* dapat digunakan sebagai flag penutup untuk satu frame dan flag pembuka untuk yang berikutnya. Di kedua sisi *user-network interface*, penerima terus mencari urutan bendera untuk disinkronkan pada awal frame. Saat menerima *frame*, stasiun terus mencari urutan itu untuk menentukan akhir *frame*. Karena protokol memungkinkan adanya pola bit yang berubah-ubah yaitu, tidak ada batasan pada konten dari berbagai bidang yang diberlakukan oleh link protokol, tidak ada jaminan bahwa pola 01111110 tidak akan muncul di suatu tempat di dalam frame, sehingga menghancurkan sinkronisasi.

Untuk menghindari masalah ini, prosedur yang dikenal sebagai *bit stuffing*. Untuk semua bit antara flag awal dan akhir, pemancar menyisipkan 0 bit ekstra setelah setiap kemunculan lima 1 dalam *frame*. Setelah mendeteksi *flag* awal, penerima memonitor *bit stream*. Ketika pola lima 1 muncul, bit keenam diperiksa. Jika bit ini 0, itu dihapus. Jika bit keenam adalah 1 dan bit ketujuh adalah 0, kombinasi diterima sebagai flag. Jika bit keenam dan ketujuh sama-sama 1, pengirim mengindikasikan kondisi gagal. Dengan penggunaan *bit stuffing*, pola bit sewenang-wenang dapat dimasukkan ke dalam bidang data frame, ini dikenal sebagai *data transparency*<sup>[23]</sup>.

### 2.2.9.1 *Flow Control*

*Flow Control* merupakan teknik untuk memastikan bahwa entitas yang mentransmisikan tidak melebihi entitas penerima dengan data. Terdapat dua *flow control* yaitu *Stop-and-wait Flow Control* dengan *Sliding Window Flow Control*.

#### 1. *Stop-and-wait Flow Control*

Bentuk paling sederhana dari aliran kontrol dikenal sebagai *stop-and-wait flow control*, cara kerjanya sebagai berikut. Entitas sumber mentransmisikan *frame*. Setelah entitas tujuan menerima *frame*, itu menunjukkan kesediaannya untuk menerima *frame* lain dengan mengirimkan kembali pemberitahuan ke *frame* yang baru saja diterima. Pengirim harus menunggu sampai menerima *acknowledgment* sebelum mengirim *frame* berikutnya. Dengan demikian tujuan dapat menghentikan aliran data hanya dengan menahan pengakuan. Prosedur ini berfungsi dengan baik dan, memang, hampir tidak dapat diperbaiki ketika pesan dikirim dalam beberapa *frame* besar. Namun, sering kali suatu sumber akan memecah blok data yang besar menjadi blok yang lebih kecil dan mengirimkan data dalam banyak *frame*. Ini dilakukan karena alasan ukuran *buffer receiver* terbatas. Semakin lama waktu pengiriman, semakin besar kemungkinan akan adanya kesalahan, yang mengharuskan pengiriman ulang seluruh *frame*. Dengan *frame* yang



lebih kecil, kesalahan terdeteksi lebih cepat, dan jumlah data yang lebih kecil perlu dikirim ulang<sup>[23]</sup>.

## 2. *Sliding Window Flow Control*

Inti dari masalah yang diuraikan sejauh ini adalah bahwa hanya satu *frame* pada satu waktu yang dapat transit. Dalam situasi di mana panjang bit link lebih besar dari panjang *frame*, hasil ketidakefisienan. Efisiensi dapat sangat ditingkatkan dengan memungkinkan beberapa *frame* dalam perjalanan pada saat yang sama. Sebagai contoh dua stasiun, A dan B, terhubung melalui full-duplex link. Stasiun B mengalokasikan ruang penyangga untuk *frame* W. Dengan demikian, B dapat menerima *frame* W, dan A diizinkan untuk mengirim *frame* W tanpa menunggu *acknowledged* apapun. Untuk melacak *frame* mana yang telah diakui, masing-masing diberi label dengan nomor urut. B mengakui bingkai dengan mengirimkan pemberitahuan yang mencakup nomor urut dari *frame* berikutnya yang diharapkan. *acknowledged* ini juga secara implisit mengumumkan bahwa B siap untuk menerima W *frame* berikutnya, dimulai dengan nomor yang ditentukan. Skema ini juga dapat digunakan untuk mengakui beberapa *frame*<sup>[23]</sup>.

### 2.2.9.2 *Error Control*

Mengacu pada mekanisme untuk mendeteksi dan memperbaiki kesalahan yang terjadi dalam transmisi *frame*. Mekanisme ini disebut dengan *Automatic Repeat Request* (ARQ) efek dari ARQ adalah untuk mengubah data link yang *unreliable* menjadi *datalink* yang *reliable*. Terdapat 3 versi ARQ yang sudah di standarisasi yaitu *Stop-and-Wait* ARQ, *Go-Back-N* ARQ, dan *Selective-reject* ARQ.

#### 1. *Stop and Wait* ARQ

Stasiun pengirim mentransmisikan satu *frame* dan kemudian harus menunggu *acknowledgment* (ACK). Tidak ada *frame* data lain yang dapat dikirim sampai balasan stasiun tujuan tiba di stasiun sumber. Dua macam error yang bisa terjadi. Pertama, *frame* yang datang di

tempat tujuan bisa rusak. Penerima mendeteksi ini dengan menggunakan teknik *error detection* dan hanya membuang *frame*. Untuk menjelaskan kemungkinan ini, *transmitter* dilengkapi dengan *timer*. Setelah *frame* ditransmisikan, *transmitter* menunggu untuk *acknowledgment*. Jika tidak ada *acknowledgment* yang diterima pada saat *timer* berakhir, maka *frame* yang sama dikirim lagi. Metode ini mensyaratkan *transmitter* menyimpan salinan *frame* yang ditransmisikan sampai penerimaan diterima untuk *frame* itu.

Jenis kesalahan kedua adalah *acknowledgment* yang rusak.. Stasiun A mengirim *frame*. *Frame* diterima dengan benar oleh stasiun B, yang merespons dengan *acknowledgment* (ACK). ACK rusak dalam perjalanan dan tidak dapat dikenali oleh A, yang, oleh karena itu, akan habis dan mengirim ulang *frame* yang sama. *Frame* duplikat ini tiba dan diterima oleh B. Oleh karena itu, B menerima dua salinan dari *frame* yang sama seolah-olah mereka terpisah. Untuk menghindari masalah ini, *frame* diberi label bergantian dengan 0 atau 1, dan ucapan terima kasih positif dalam bentuk ACK0 dan ACK1. Sesuai dengan konvensi jendela geser, ACK0 mengakui penerimaan bingkai nomor 1 dan menunjukkan bahwa penerima siap untuk bingkai bernomor 0<sup>[23]</sup>.

## 2. Go-Back ARQ

Bentuk error control berdasarkan jendela aliran kontrol geser yang paling umum digunakan disebut go-back-N ARQ. Dalam metode ini, *transmitter* dapat mengirim serangkaian *frame* berurutan modulo beberapa nilai maksimum dengan menggunakan *sliding-window flow control technique*. Meskipun tidak ada kesalahan yang terjadi, tujuan akhir akan mengakui *frame* masuk seperti biasa (RR = *receive ready*). Jika stasiun tujuan mendeteksi *error* dalam sebuah *frame*, mungkin stasiun akan mengirim *acknowledgment* negatif (REJ = *reject*) untuk *frame* itu, seperti dijelaskan dalam aturan berikut. Stasiun tujuan akan

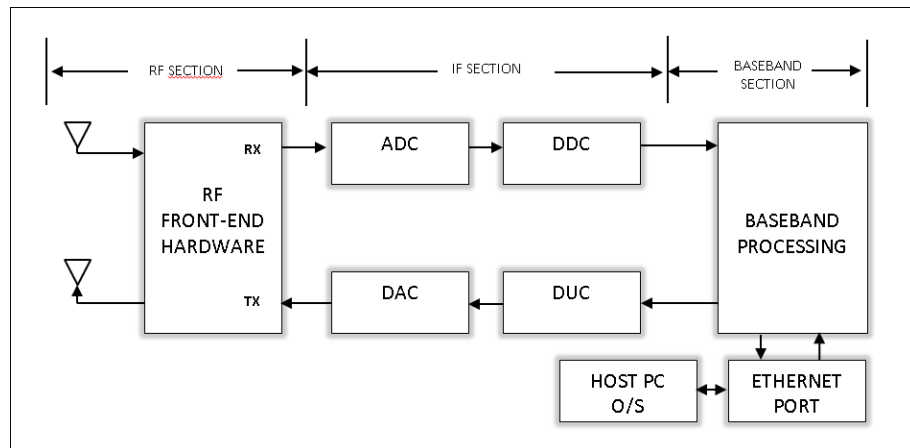
membuang *frame* itu dan semua *frame* yang masuk di masa depan sampai *frame* dalam kesalahan diterima dengan benar. Dengan demikian, *reciever*, ketika menerima REJ, harus mentransmisikan ulang *frame* dalam *error* ditambah semua *frame* berikutnya yang ditransmisikan untuk sementara.

### 3. Selective-Reject ARQ

Dengan ARQ yang ditolak secara selektif, satu-satunya *frame* yang dikirim ulang adalah mereka yang menerima *acknowledgment* negatif, dalam hal ini, disebut SREJ, atau mereka yang *time out*. Selective reject tampaknya lebih efisien daripada go-back-N karena meminimalkan jumlah pengiriman ulang. Di sisi lain, penerima harus memelihara buffer yang cukup besar untuk menyimpan *frame post* SREJ sampai *frame* yang salah dikirimkan kembali dan harus mengandung logika untuk memasukkan kembali *frame* tersebut dalam urutan yang benar. *Transmitter* juga membutuhkan logika yang lebih kompleks untuk dapat mengirim keluar dari urutan. Karena komplikasi seperti itu, ARQ select-reject jauh lebih jarang digunakan daripada AR-go-back-N. Penolakan selektif adalah pilihan yang berguna untuk link satelit karena keterlambatan propagasi yang lama<sup>[23]</sup>.

## 2.3 *Software-Defined Radio*

*Software Defined Radio* (SDR) merupakan perangkat sistem komunikasi radio yang dapat diatur secara pemrograman menggunakan *software*. Beberapa fungsi modul dalam sistem radio seperti modulasi, pembangkit sinyal, coding dan protocol dapat diimplementasikan menggunakan perangkat lunak yang terintegrasi dengan SDR tersebut.

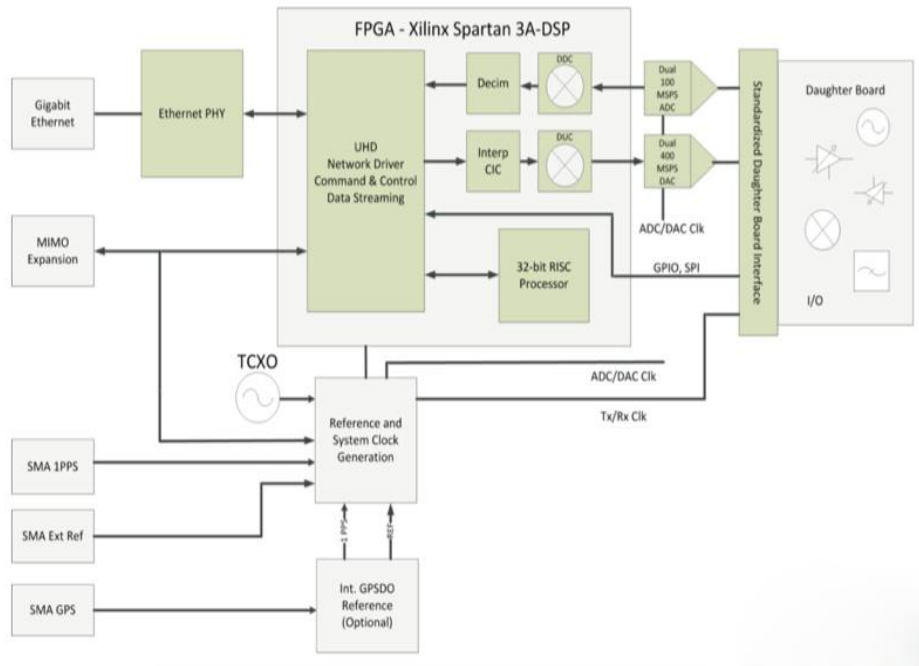


Gambar 2.21 Blok Diagram SDR

Gambar 2.21 merupakan arsitektur SDR, dimana *RF Front-end* pada receiver berfungsi menerima signal dari pemancar dan dikuatkan menggunakan *low noise amplifier*, kemudian dilakukan *down conversion* dari frekuensi RF ke IF. Sedangkan pada transmitter, *RF Frond-end* berfungsi sebagai *Up conversion* dari frekuensi IF ke RF kemudian melakukan penguatan sinyal RF menggunakan *high power amplifier*. Blok ADC/DAC berfungsi untuk melakukan konversi sinyal analog ke digital atau sinyal digital ke analog. Pada jalur penerima pada arsitektur tersebut terdapat rangkaian *Digital Down Converter* (DDC) yang berfungsi mengubah frekuensi sinyal digital yang diterima menjadi frekuensi IF sinyal *baseband* kompleks yang berpusat pada frekuensi 0 Hz. Sedangkan pada jalur pemancar terdapat rangkaian *Digital Up Converter* (DUC). Perangkat SDR diprogram melalui *firmware* dan *driver hardware* yang dapat dimuat dan diperbaharui bila terkoneksi dengan host personal computer (PC).

Salah satu contoh SDR adalah perangkat *Universal Software Radio Peripheral* (USRP). USRP merupakan suatu prototipe perangkat radio yang diproduksi oleh *National Instrument* (NI) yang memiliki beberapa jenis dengan fungsi yang berbeda-beda. USRP terdiri dari *motherboard* dan *daughterboard*. Penentuan penggunaan dari jenis *daughterboard*, dapat disesuaikan dengan frekuensi kerja yang akan diinginkan. USRP merupakan perangkat yang dikontrol oleh software LabVIEW untuk

pemrosesan sinyal. Untuk diagram blok yang sederhana dari USRP dapat dilihat pada gambar 2.22. Sedangkan sebagai contoh untuk pemilihan perangkat berdasarkan frekuensi, USRP memiliki beberapa *daughterboard* untuk penggunaannya seperti pada table 2.12.



Gambar 2.22 Struktur Bagan USRP N210

Tabel 2.12 Pembagian frekuensi untuk *daughterboard*

<i>Daughterboard</i>	<i>Frequency Coverage</i>
WBX-120	50 MHz - 2.2 GHz
SBX-120	400 MHz - 4.4 GHz
CBX-120	1.2 GHz - 6 GHz
UBX-120	10 MHz - 6 GHz
WBX	50 MHz - 2.2 GHz
SBX	400 MHz - 4.4 GHz
CBX	1.2 GHz - 6 GHz
UBX-40	10 MHz - 6 GHz
TVRX2	50 MHz - 860 MHz
DSBRX2	800 MHz - 2.3 GHz
BasicRX / BasicTX	1 - 250 MHz
LFRX / LFTX	DC - 30 MHz

## 2.4 Antena Dipole

Antena adalah perangkat konduktor yang digunakan untuk memancarkan gelombang elektromagnetik menuju ruang bebas atau menangkap gelombang elektromagnetik dari ruang bebas. Pada pemancar energi listrik dari pemancar diubah menjadi gelombang elektromagnetik kemudian dipancarkan ke udara bebas. Pada sisi penerima gelombang elektromagnetik dari udara bebas diterima oleh antenna dan diubah menjadi energi listrik. Antena merupakan perangkat perantara antara media kabel dan udara, sehingga antena harus mempunyai sifat yang sesuai dengan media kabel pencatunya.

Gelombang elektromagnetik yang diradiasikan oleh antena terdiri dari dua komponen yaitu medan listrik dan medan magnetik. Dimana energi total tersebut dipancarkan dalam bentuk gelombang yang hampir konstan ke udara bebas dan ada beberapa yang terserap oleh tanah. Antena dikatakan baik apabila dapat memancarkan gelombang elektromagnetik ke ruang bebas dengan efisiensi yang tinggi, dan dapat mengarahkan sebagian besar daya radiasinya ke arah yang diinginkan dan menekan radiasi ke arah yang lain.

Antena dengan pancaran gelombang elektromagnetik yang memiliki nilai efisiensi tinggi dapat dicapai apabila sebuah antena memiliki tingkat penyesuaian yang baik antara saluran transmisi dan ruang bebas, atau impedansi saluran transmisi sistem *transceiver* “*matched*” dengan impedansi radiasi antena tersebut. Dalam hal ini antena disebut sebagai *matching device*, yang merupakan fungsi dari sebuah antena. Antena juga memiliki fungsi yang lain yaitu sebagai *converter* karena antenna tersebut mengubah dari sinyal listrik menjadi sinyal elektromagnetik atau sebaliknya. Antena juga bisa berfungsi sebagai radiator karena antenna tersebut memancarkan gelombang elektromagnetik ke udara bebas, dan dikatakan *re-radiator* jika antenna menerima radiasi gelombang elektromagnetik dari udara bebas.

Salah satu ciri antena dapat dikatakan baik jika mempunyai nilai parameter antena seperti *voltage standing wave ratio* (VSWR) kurang dari

2. VSWR adalah perbandingan antara amplitude gelombang berdiri (standing wave) maksimum ( $|V|_{max}$ ) dengan minimum ( $|V|_{min}$ ). Pada saluran transmisi ada dua komponen gelombang tegangan, yaitu tegangan yang dikirimkan ( $V_0^+$ ) dan tegangan yang direfleksikan ( $V_0^-$ ). Perbandingan antara tegangan yang direfleksikan dengan yang dikirimkan disebut sebagai koefisien refleksi tegangan ( $\Gamma$ )<sup>[24]</sup>:

$$\Gamma_L = \frac{e_1^-}{e_1^+} = \frac{Z_1 - Z_m}{Z_1 + Z_m} \quad (2.28)$$

Dimana:

$$\begin{aligned} e_1^- &= \text{tegangan pantul} & \Gamma_L &= \text{impedansi beban} \\ e_1^+ &= \text{tegangan datang} & Z_{in} &= \text{impedansi input} \end{aligned}$$

Pada beberapa contoh kasus, ketika bagian imajiner adalah nol, maka :

- $\Gamma = -1$  ; refleksi negatif maksimum, ketika saluran terhubung singkat
- $\Gamma = 0$  ; tidak ada refleksi, ketika saluran dalam keadaan matched sempurna.
- $\Gamma = +1$  ; refleksi positif maksimum, ketika saluran dalam rangkaian terbuka

Sehingga VSWR, dapat dinyatakan sebagai berikut:

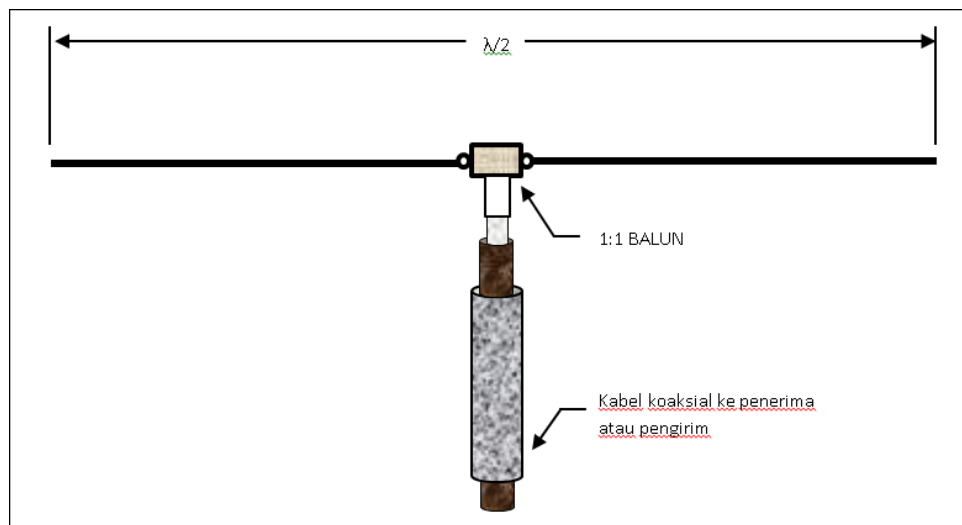
$$VSWR = \frac{|\tilde{V}|_{max}}{|\tilde{V}|_{min}} = \frac{1 + |\Gamma|}{1 - |\Gamma|} \quad (2.29)$$

Dimana :

$$\begin{aligned} VSWR &= \text{voltage standing wave ratio} \\ |\tilde{V}|_{max} &= \text{amplitudo } \textit{standing wave} \text{ maksimum} \\ |\tilde{V}|_{min} &= \text{amplitudo } \textit{standing wave} \text{ minimum} \\ \Gamma &= \text{koefisien refleksi tegangan} \end{aligned}$$

Antena *dipole* adalah suatu antena resonan yang memiliki panjang antena yaitu sebesar setengah lamda. Antena *dipole* sebenarnya merupakan antena yang dibuat dari kawat tembaga dan yang memiliki panjang ukuran tertentu agar beresonansi pada frekuensi kerja yang diinginkan. Antena *dipole* yang hanya terdiri dari satu kawat saja disebut

*single wire dipole*, sedangkan antenna *dipole* yang terdiri dari dua kawat yang ujung-ujungnya dihubungkan dinamakan *two wire folded dipole*. Pada antenna *dipole* diperlukan sebuah *balance to unbalance* (balun) jika feeder yang digunakan berupa kabel koaksial. Balun tersebut berfungsi sebagai *matching impedance*, dimana impedansi antenna *dipole* adalah sebesar  $(73+j42,5) \Omega$  sedangkan feeder yaitu kabel koaksial memiliki impedansi sebesar  $50 \Omega$ . Sehingga kabel koaksial ini bersifat unbalance sedangkan antenna bersifat balance.



Gambar 2.23 Antena Dipole  $\frac{1}{2}$  lamda

Gambar 2.23 merupakan gambar antenna *single wire dipole* dimana terdapat satu kawat disisi kanan dan kiri. Agar antenna tersebut bekerja dengan baik maka kedua lengan kawat antenna dicatu dengan arus yang sama. Apabila antenna *dipole* langsung terhubung dengan kabel koaksial akan mengakibatkan arus pada kawat kanan tidak sama dengan pada kawat kiri. Sehingga agar antenna *dipole* bersifat balance, antenna *dipole* dicatu arus yang sama atau seimbang di sepanjang lengannya. Namun berbeda dengan karakteristik pada kabel koaksial dimana beberapa arus dapat berjalan di bagian luar koaksial, yang menyebabkan kondisi unbalance. Solusi untuk masalah ini adalah menggunakan balun yang berfungsi untuk memaksa jalur transmisi yang unbalance pada komponen balance.



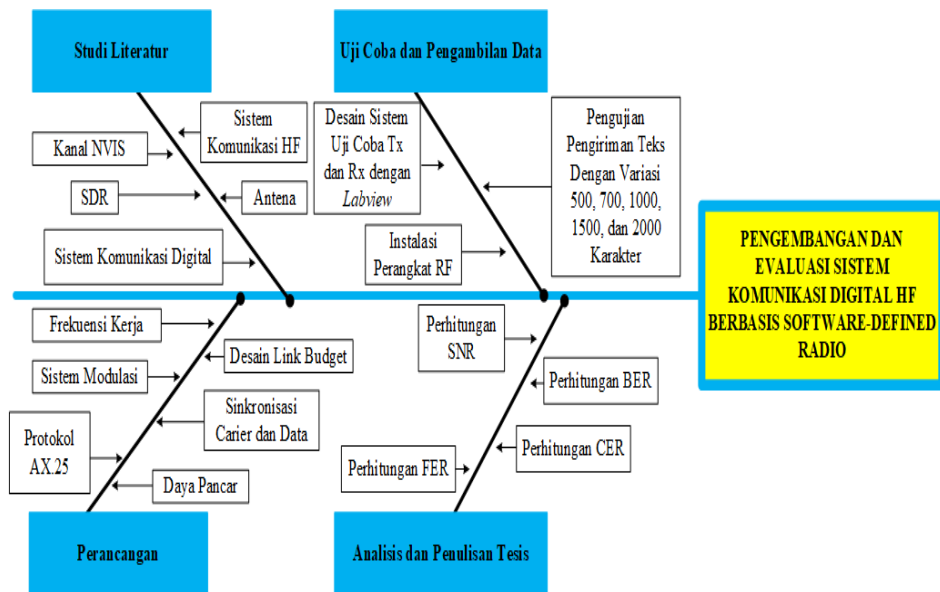
*Halaman ini sengaja dikosongkan*

## BAB 3

### METODE PENELITIAN

#### 3.1 Skema Penelitian

Pada sub bab ini penulis menguraikan desain, metode, implementasi dan uji coba pada penelitian yang dilakukan tentang Pengembangan dan Evaluasi Sistem Komunikasi Digital HF berbasis *Software-Defined Radio*, yang sudah diringkas dalam diagram *fishbone*. Pada diagram *fishbone* diuraikan tahapan-tahapan secara rinci dimulai dari tahap studi literatur, perancangan, uji coba, pengambilan data, analisis dan penulisan tesis. Secara ringkas skema penelitian (*fishbone*) ditunjukkan pada gambar 3.1.



Gambar 3.1 Skema Penelitian (*fishbone*)

Pada skema penelitian ini dilakukan dalam beberapa tahapan. Pada tahap awal dilakukan studi literatur yang meliputi beberapa dasar teori yang mendukung dalam proses penelitian diantaranya yaitu tentang sistem komunikasi HF, sistem link komunikasi digital, kanal NVIS, SDR dan Antena. Tahapan kedua yaitu melakukan perancangan sistem yang meliputi proses penentuan frekuensi kerja, desain *link budget*, sistem modulasi, protokol

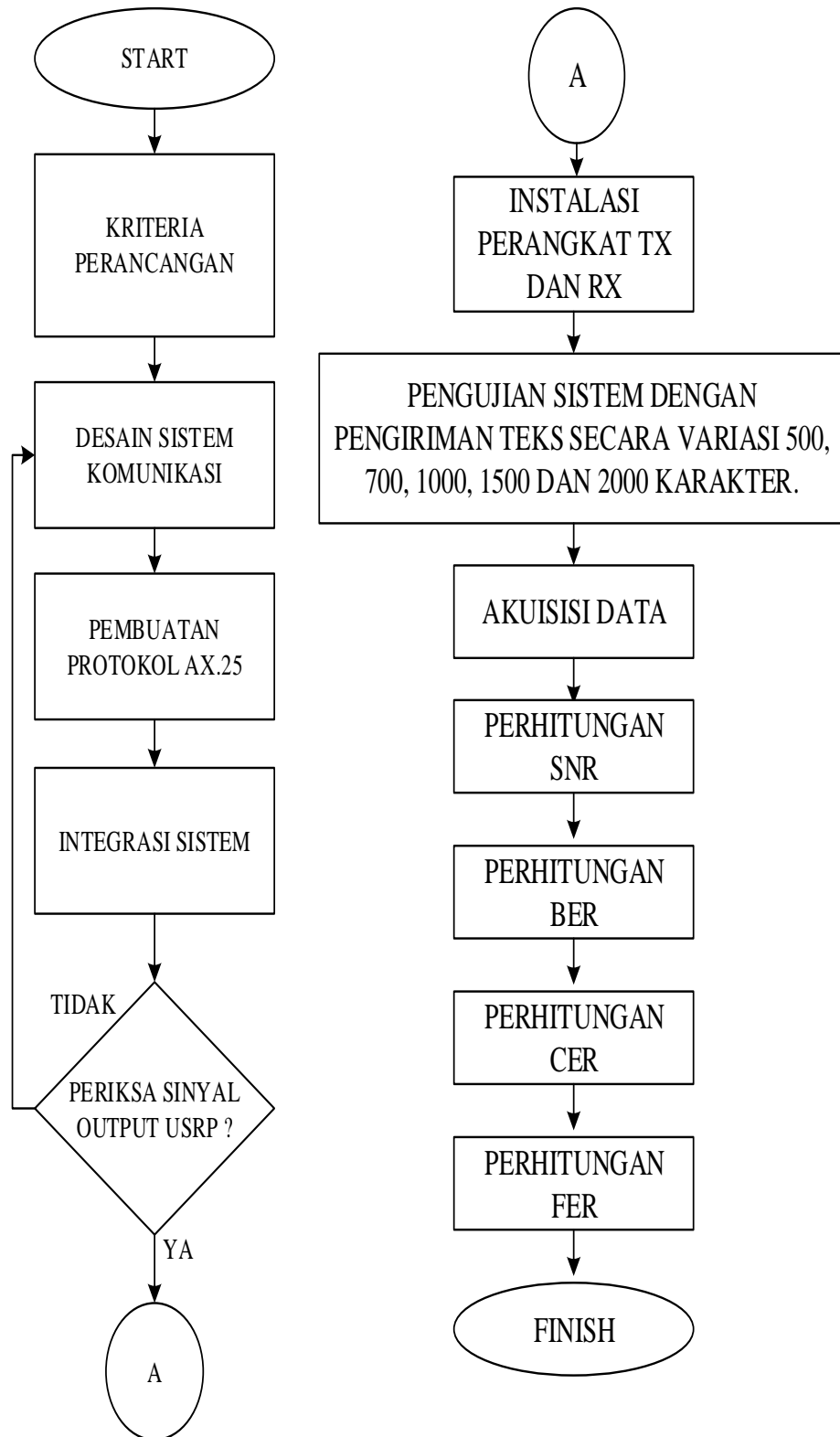
AX.25, teknik sinkronisasi carier dan data, , dan daya pancar. Tahapan ketiga adalah uji coba sistem dan pengambilan data, pertama desain sistem uji coba Tx dan Rx dengan menggunakan *software* simulasi *Labview*, pada proses simulasi menggunakan *software labview* dapat di peroleh parameter-parameter yang dibutuhkan pada saat melaksanakan uji coba di lapangan dengan sebelumnya melakukan instalasi perangkat RF dan proses pengujian pengambilan data dengan melakukan pengiriman teks secara variasi dari 500, 700, 1000, 1500, dan 2000 karakter. Setelah dilakukan pengambilan data selanjutnya dilakukan perhitungan SNR, BER, CER dan SNR serta dilakukan analisis. Selanjutnya ditulis dalam draft paper dan submit conference serta membuat laporan akhir penelitian.

Untuk pengujian sistem dilakukan antara antenna jembatan gedung AJ dengan antena gedung B pada laboratorium antenna dan propagasi Institut Teknologi Sepuluh Nopember selama 7 hari.

### **3.2 Diagram Alir Penelitian**

Pada penelitian ini, untuk menyelesaikan dengan hasil yang baik dan signifikan dalam desain sistem komunikasi digital HF berbasis Software-Defined Radio diperlukan beberapa tahapan yang perlu dilakukan dan diselesaikan dengan baik untuk mendapatkan hasil yang signifikan. Beberapa tahapan diantaranya kriteria perancangan, desain sistem komunikasi, pembuatan modul protocol AX.25, integrasi sistem, dan periksa sinyal output USRP. Jika sinyal output USRP sdh berhasil diterima maka dilanjutkan ke proses selanjutnya, dan jika tidak maka harus diperbaiki lagi desain sistem komunikasinya. Setelah di perika diterima dengan baik maka dilanjutkan ke instalasi perangkat pemancar dan penerima. Ketika sudah di instalasikan dilanjutkan dengan pengujian sistem dengan pengiriman teks secara variasi dari 500, 700, 1000, 1500, dan 2000 karekter. Setelah berhasil data yang di ambil kemudian di akuisi dan selanjutnya bisa di olah untuk penghitungan SNR, BER, CER dan FER. Untuk dapat mengetahui tahapan – tahapan di setiap proses pada penelitian

ini, maka penulis membuat secara ringkas dan detail pada diagram alir seperti ditunjukkan pada gambar 3.2 yaitu diagram alir penelitian.



Gambar 3.2 Diagram Alir Penelitian

### 3.3 Rancangan Penelitian

Pada penelitian pengembangan dan evaluasi sistem komunikasi digital HF berbasis *Software-Defined Radio* menggunakan range frekuensi 3 – 30 MHz. Lokasi penelitian ini dilakukan tepatnya di kota Surabaya sebagai *transmitter* dan *receiver*. Pada *transmitter* yaitu menggunakan antena *dipole* yang diletakkan di atas jembatan Gedung AJ kampus elektro ITS dengan arah pancaran ke timur dan barat dengan polarisasi horizontal, dan. Pada sisi *Receiver* juga menggunakan satu antena *dipole* yang diletakkan di sepanjang gedung B kampus elektro ITS dengan arah pancaran ke utara dan selatan dengan polarisasi horizontal.

#### 3.3.1 Desain Frekuensi Kerja

Pada penelitian pengembangan dan evaluasi sistem komunikasi digital HF berbasis *Software-Defined Radio* pemilihan frekuensi kerja yang digunakan pada *transmitter* dan *receiver* adalah 7 MHz. Frekuensi tersebut dapat digunakan berkisar waktu antara pukul 07.00 WIB sampai dengan pukul 16.00 WIB. Pada sistem komunikasi HF, beberapa faktor yang mempengaruhi frekuensi kerja pada saat digunakan diantaranya adalah waktu pengukuran. Setelah fajar, radiasi matahari menyebabkan elektron diproduksi di ionosfer dan frekuensi meningkat dengan cepat hingga maksimum sekitar siang hari. Pada sore hari, frekuensi mulai turun karena kehilangan elektron dan lapisan D, E dan F1 menghilang <sup>[11]</sup>. Dengan demikian maka pada penelitian ini menggunakan pengujian jarak dekat sehingga komunikasi pada pagi sampai sore hari sangat relevan untuk digunakan.

#### 3.3.2 Pemilihan Lokasi Pemancar dan Penerima

Pada penelitian ini dilakukan pada dua tempat sebagai implementasi pengujian, lokasi antena pemancar berada di jembatan gedung AJ Teknik Elektro ITS dan lokasi antena penerima berada di gedung B Teknik Elektro ITS, dengan jarak  $\pm 50$  meter. Seperti ditunjukkan pada gambar 3.3 sebagai pemancar dan 3.4 sebagai penerima sebagai berikut.



Gambar 3.2 Antena Pemancar (TX) di Jembatan Gedung AJ



Gambar 3.3 Antena Penerima (RX) di Gedung B

### 3.4 Desain Sistem Keseluruhan

Pada penelitian ini perangkat *transmitter* dan *receiver* terdapat masing-masing 1 *Universal Software Radio Peripheral* (USRP) tipe N210 yang diintegrasikan dengan *software labview*. Perangkat USRP dan *software labview* tersebut digunakan untuk desain, implementasi, pengujian dan pengambilan data dari antena yang nantinya akan diolah menggunakan *software Matrix Laboratory* (Matlab) untuk mendapatkan nilai *Signal to Noise Ratio* (SNR), *Bit Error Rate* (BER), *Character Error Rate* (CER) dan *Frame Error Rate* (FER). Pengiriman data yang dilakukan pada penelitian ini menggunakan pengiriman data teks.

Proses yang terjadi pada sistem komunikasi sisi pemancar adalah dimana deretan bit data yang dikirimkan untuk proses transmisi data pada sistem komunikasi sebelumnya dilakukan *mapping* terlebih dahulu. Arti

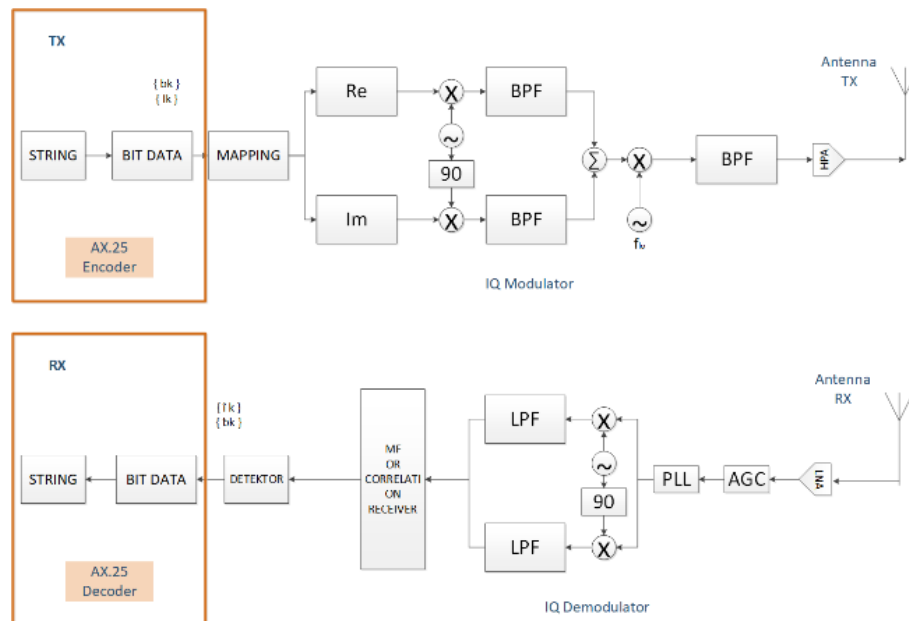
dari *mapping* disini adalah untuk mengubah deretan bit menjadi sinyal LPE (*Low Pass Equivalent*) yang nantinya dapat di kelompokkan ke dalam dua sinyal yaitu Re (*real*) dan Im (*Imaginner*). Proses tersebut terjadi pada *software labview* yang sudah terinstall di laptop. Setelah dari proses pemisahan sinyal tersebut, maka data di jadikan satu ke dalam IQ *Modulator* dimana untuk perangkat terdapat di perangkat *Motherboard* USRP. Dengan memiliki sistem kerja berupa dua sinyal yang masuk di kuatkan dan disesuaikan dengan kondisi kanal menggunakan *oscillator* dengan mengikuti pergerakan fasa hingga 90 derajat.

Setelah dua sinyal itu masuk maka diubah kembali menjadi sinyal BPF (*Band Pass Filter*). Kemudian kedua sinyal yang *Real* dan *Imaginer* di jumlahkan terlebih dahulu untuk dapat melihat kualitas sinyalnya. Setelah dapat melihat kualitas sinyalnya dan dirasa kurang, maka dapat di kuatkan kembali dengan *oscilator* sebagai penguat berdasarkan frekuensinya masing-masing. Sebelum dilakukan transmisi data pada sistem komunikasi maka sinyal tersebut kembali di rubah ke dalam Sinyal BPF (*Band Pass Filter*) dan dikuatkan kembali oleh HPA (*High Power Amplifier*) untuk dapat menembus kondisi di ionosfer. Setelah itu maka transmisi pengiriman data dapat di lakukan melalui antena pada sisi pemancar.

Sedangkan, proses yang terjadi pada sistem komunikasi sisi penerima adalah antena menerima sinyal dari kanal. Sinyal yang diterima tersebut adalah sinyal dalam kondisi yang kurang baik yang artinya, kondisi dimana sinyal telah melalui transmisi berupa media udara yang banyak terjadi gangguan (*noise*), maka perlu di kuatkan kembali dengan perangkat keras bernama LNA (*Low Noise Amplifier*) yang berguna untuk menguatkan kualitas sinyal dan dapat membuat data yang dikirimkan pada proses pengiriman dari pemancar dapat diterima pada sisi penerima. Untuk meningkatkan sinyal maka diperlukan AGC (*Automatic Gain Control*) yang memiliki fungsi untuk menguatkan sinyal yang sangat lemah agar *gain* antena yang diterima dapat berfungsi maksimal dengan kualitas sinyal yang konstan. Selain itu AGC dapat mengkompensasi suatu redaman yang biasanya terjadi pada perangkat USRP (*daughterboard*). Untuk sinkronisasi

gelombang pembawa dalam proses demodulasi diperlukan PLL (*Phase-Locked Loop*). PLL memiliki fungsi untuk menghilangkan perbedaan sudut fasa atau frekuensi dalam proses demodulasi. Kemudian pada *IQ modulator*, sinyal yang diterima pada penerima di kelompokkan menjadi dua yaitu *Real* dan *Imaginer* dengan dikuatkan kembali dengan gelombang pembawa. Setelah dari proses tersebut, sinyal akan diubah menjadi LPF (*Low Pass Filter*), dimana prosesnya terjadi pada *motherboard*.

Penelitian ini juga di desain dengan menggunakan *protocol AX.25* pada sisi pemancar dan sisi penerima. Program *protocol AX.25* di integrasikan pada keseluruhan sistem pemancar dan penerima. *Protocol AX.25* sebagai *encoder* terletak di pemancar dengan penempatan sebelum proses modulator dimana bit data yang dihasilkan oleh *encoder* menjadi masukan pada proses modulasi. *Protocol AX.25* sebagai *decoder* terletak di penerima dengan penempatan setelah proses demodulator terjadi. Gambar 3.4 merupakan desain pemancar dan penerima secara keseluruhan pada sistem komunikasi digital HF.

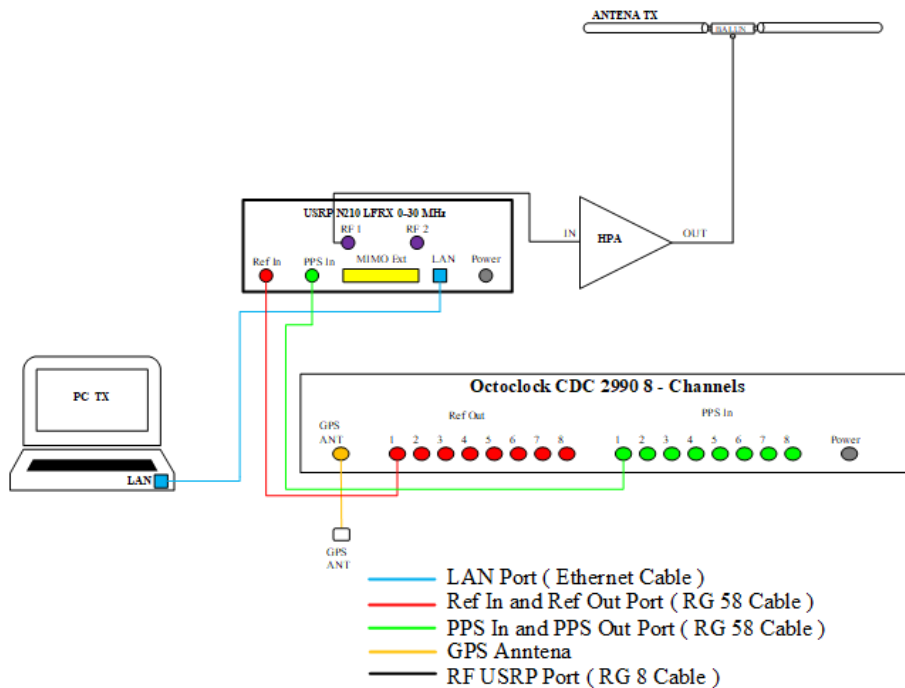


Gambar 3.4 Desain Sistem Komunikasi Digital HF Keseluruhan



### 3.4.1 Desain Sistem Pemancar

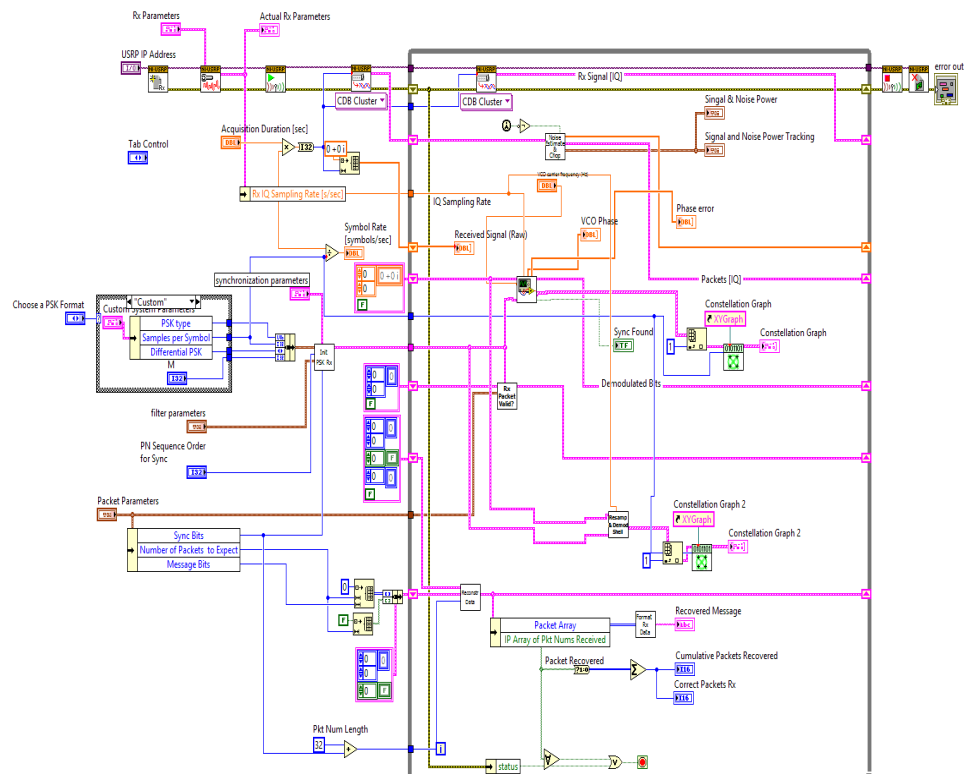
Pada desain sistem pemancar berfungsi untuk mengirimkan data teks yang bekerja pada frekuensi 7 MHz. Gambar 3.5 merupakan desain pemancar yang digunakan pada sistem ini. Pada desain pemancar terdapat perangkat USRP yang diintegrasikan dengan *software labview* yang digunakan untuk desain, implementasi, pengukuran dan pengambilan data serta laptop sebagai perangkat pengendali sekaligus perangkat yang digunakan untuk menjalankan sistem pada *software labview*. Perangkat *Octoclock* berfungsi untuk mengatur sinkronisasi. Pada perangkat *Octoclock* terdapat antenna *Global Positioning System* (GPS) yang pada waktu pengoperasian harus diletakkan di luar ruangan untuk mendapatkan sinyal GPS yang baik. Perangkat *High Power Amplifier* (HPA) berfungsi untuk menguatkan daya pancar sinyal dari USRP ke penerima melalui antenna.



Gambar 3.4 Desain Sistem Pemancar (TX)

### 3.4.1.1 Desain Sistem Komunikasi Digital HF Penggunaan Tanpa *Protocol* AX.25 di Pemancar Dengan *Software Labview*

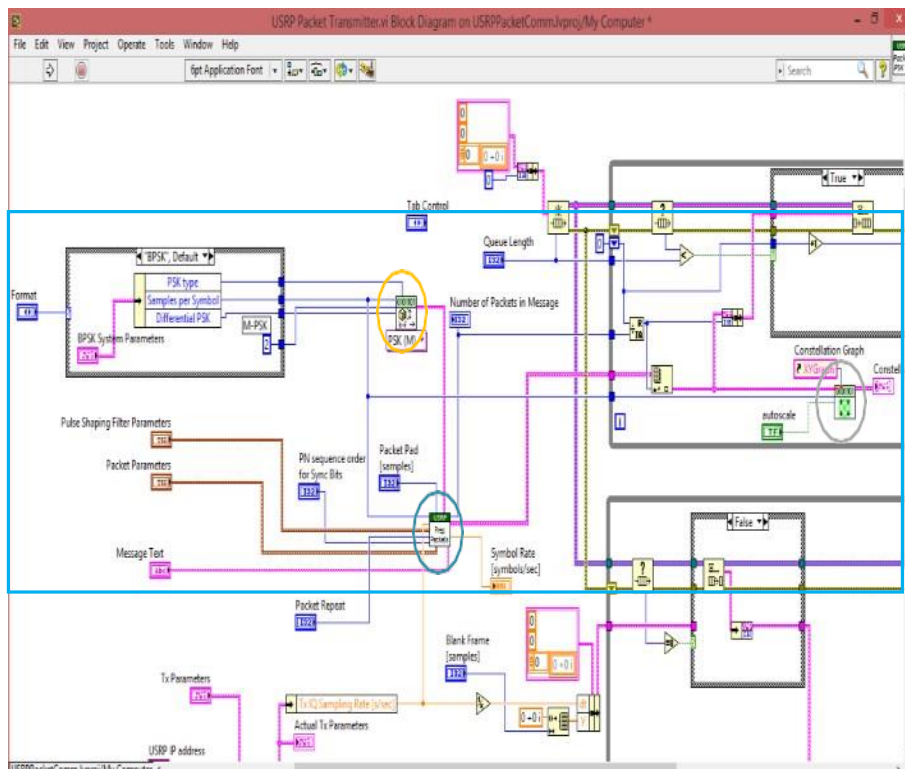
Desain sistem komunikasi digital HF tanpa *protocol* AX.25 di pemancar menggunakan *software Labview*. Gambar 3.5 merupakan tampilan desain sistem tanpa *protocol* di sisi pemancar pada *software Labview*.



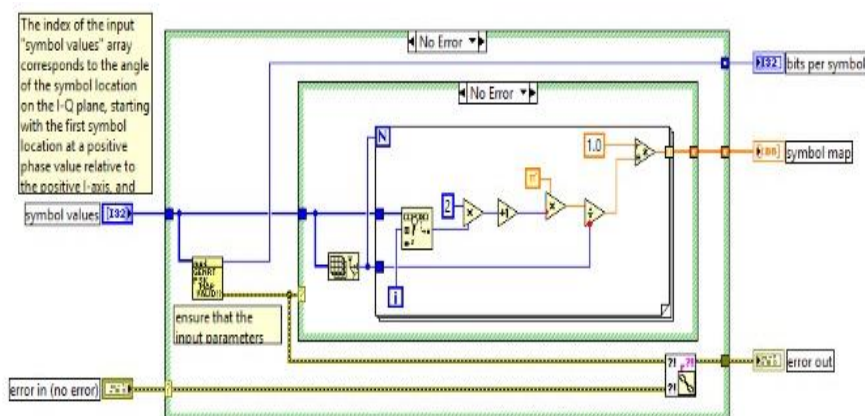
Gambar 3.5 Desain Sistem Tanpa *Protocol* di Sisi Pemancar

Pada penelitian ini di desain menggunakan modulasi tipe BPSK dengan memiliki *output* diagram konstelasi yang di tunjukan pada garis berbentuk lingkaran berwarna abu-abu yang diintegrasikan dengan sub sistem *generate system parameters* pada gambar 3.6. Gambar 3.6 merupakan gambar yang menunjukkan sub-sub *block* sisi pemancar pada *Labview* dan diberikan garis berbentuk kotak warna biru untuk menunjukan *block-block* sistem modulasi. Proses yang dilakukan pertama yaitu menentukan parameter modulasi yang digunakan. Modulasi yang digunakan pada penelitian ini adalah modulasi tipe PSK. Kemudian masuk ke sub sistem *generate system parameters* yang ditunjukkan pada garis berbentuk

lingkaran berwarna kuning. Didalam sub sistem ini terdapat validasi *system* parameter yang berfungsi untuk memastikan *input* parameter yang digunakan benar. Selanjutnya terdapat sub sistem *gray code*, dimana input dapat diproses pada sub sistem ini dan dikonversi menjadi kode biner, fungsi dari kode *gray* digunakan untuk meminimalisir kesalahan dari *input*. Desain *gray code* di dalam *sub system* dapat di tampilkan pada gambar 3.7.



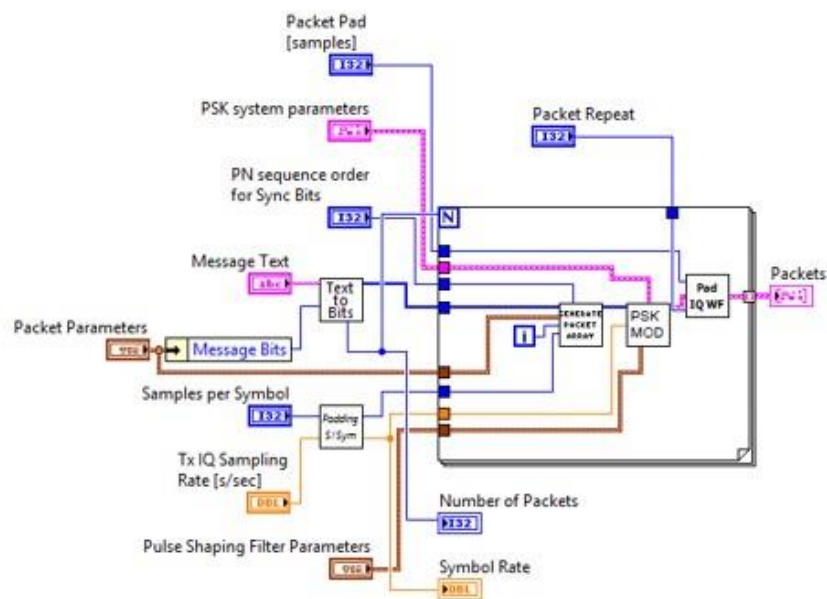
Gambar 3.6 Tampilan Sub-Sub *Block* Sistem Modulasi



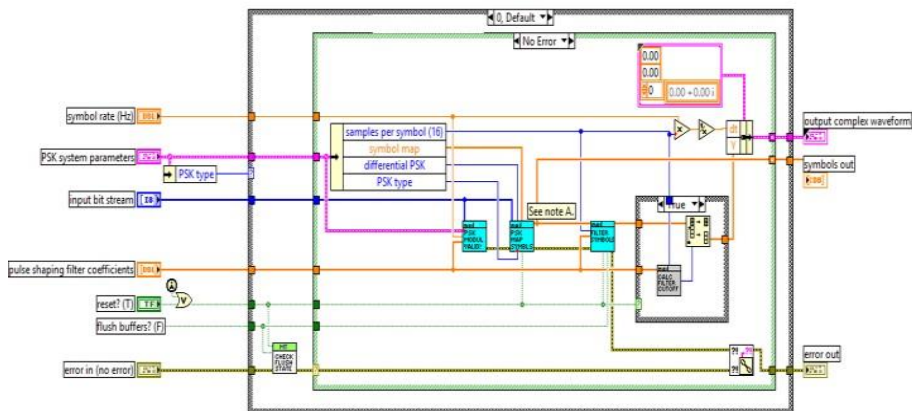
Gambar 3.7 Desain *Gray Code* Di Dalam Sub System

Selanjutnya masuk ke dalam sub sistem *generate symbol map*. Di sub sistem ini terdapat rangkaian sistem modulasi yang dimana *output* dapat melakukan *mapping* terhadap *bit rate*. Pada tahapan ini merupakan tahap awal untuk konfigurasi sistem modulasi. Setelah melakukan inialisasi awal, maka selanjutnya masuk ke dalam sub Generate Packets yang ditunjukkan pada garis berbentuk lingkaran berwarna biru pada gambar 3.6. Sub sistem ini merupakan sub sistem yang terintegrasi oleh *input* karakter, parameter modulasi, *pulse shaping*, dan *generate packet array*. Output dari sub sistem ini berupa indicator sebuah paket-paket yang akan dikirim seperti *symbol rate*, dan jumlah paket yang dikirim.

Gambar 3.7 merupakan tampilan sub PSK MOD dimana memiliki fungsi untuk proses modulasi dengan parameter-parameter yang sudah di desain di awal. Pada sistem ini didalamnya terdapat *cluster filter shaping* dan proses modulasi. Selanjutnya, dari output sub sistem sebelumnya akan menjadi input *sub-sistem* konstelasi. Sedangkan gambar 3.8 merupakan tampilan desain modulasi. Dimana modulasi BPSK merupakan bentuk dari modulasi PSK yang mana 1 bit dimodulasikan menjadi 1 simbol yang menggunakan 2 fase berbeda dan 2 simbol berbeda.



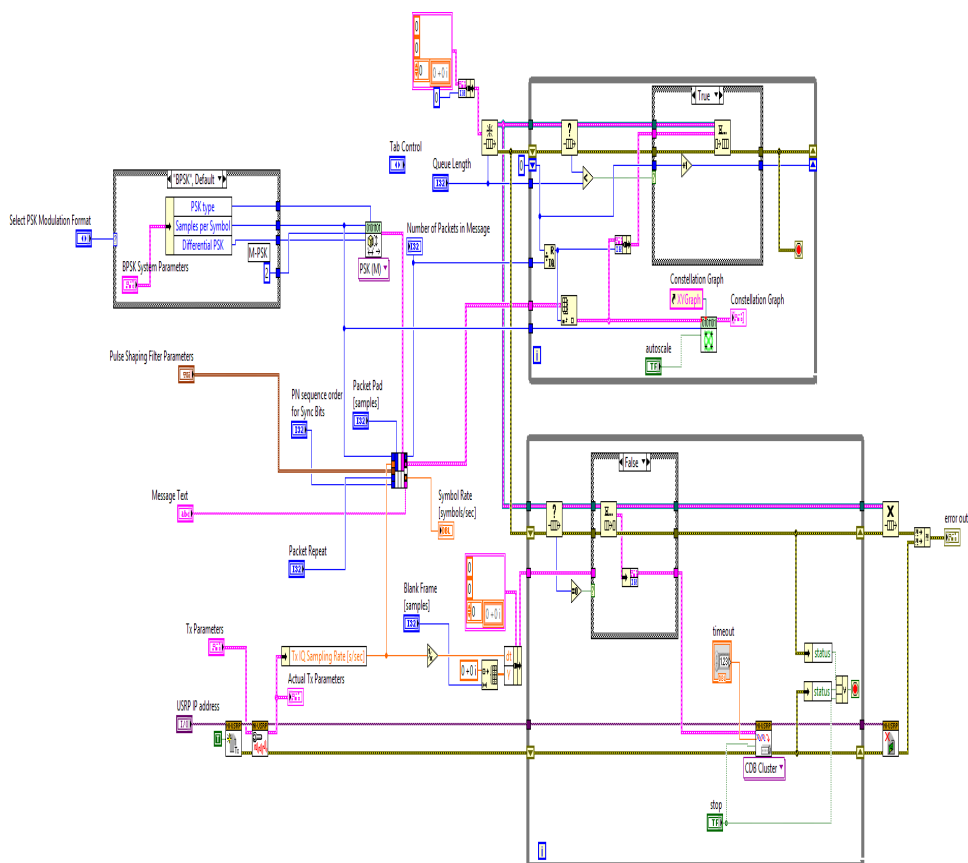
Gambar 3.8 Desain sub sistem Packet Generate



Gambar 3.9 Desain Modulasi

### 3.4.1.2 Desain Sistem Komunikasi Digital HF Penggunaan Dengan *Protocol AX.25* di Pemancar Dengan *Software Labview*

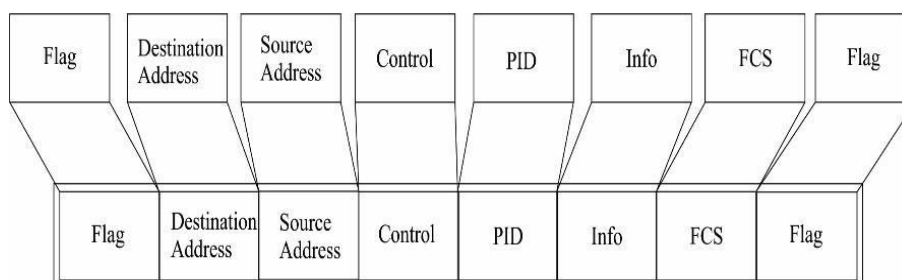
Sistem komunikasi digital HF pada *software labview* di pemancar terdapat pada gambar 3.10 dengan desain menggunakan *protocol AX.25*.



Gambar 3.10 Desain Sistem Dengan *Protocol* di Sisi Pemancar

*Protocol AX.25* terdiri dari program yang mampu menyusun informasi ke dalam bentuk frame AX.25 dari masing-masing *field* dijadikan

satu dalam satu *frame protocol AX.25*. Pada gambar 3.11 merupakan desain *frame protocol AX.25*.



Gambar 3.11 Desain *Frame Protocol AX.25*

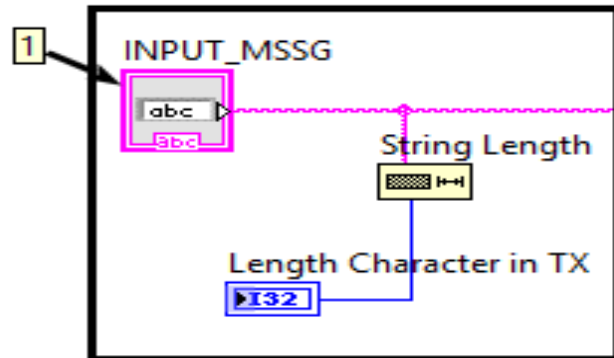
Pada masing-masing *field* diisi dengan data berupa bilangan biner dengan susunan sesuai dengan desain *frame protocol AX.25*. Desain *subfield* terdapat pada tabel 3.1 sebagai berikut.

Tabel 3.1 Desain Isi *Subfield*

<i>Octet</i>	<i>ASCII</i>	<i>Bin Data</i>	<i>Hex Data</i>
<i>Flag</i>		0111110	7E
A1	Y	01011001	59
A2	G	01000111	47
A3	3	00110011	03
A4	E	01000101	45
A5	G	01000111	47
A6	Y	01011001	59
A7	SSID	11100000	E0
A8	A	01000001	41
A9	0	00110000	30
A10	0	00110000	30
A11	0	00110000	30
A1	Y	01011001	59
A2	G	01000111	47
A12	0	00110000	30
A13	2	00110010	32
A14	SSID	01100001	61
<i>Control</i>	1	00111110	3E
<i>PID</i>	None	11110000	F0
<i>FCS</i>	Part1	10011000100010101	13115
<i>FCS</i>	Part2	10011000100010101	13115
<i>Flag</i>	Part2	01111110	7E

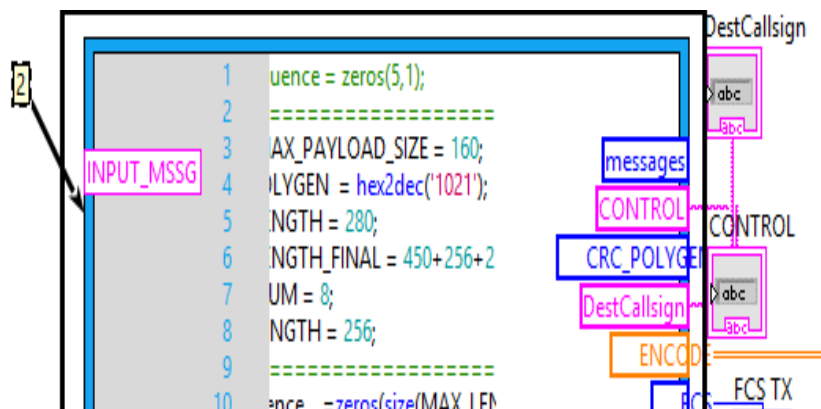
Proses pertama yang dilakukan adalah memberikan *input* berupa teks atau karakter dimana format pada *labview* menggunakan indikator *String*

seperti yang terlihat pada gambar 3.12. Selain itu, ditambahkan fungsi “*String Length*” untuk menghitung panjang jumlah karakter (*byte*) dalam *string*.



Gambar 3.12 Desain input teks pada *software labview*

Lalu langkah kedua adalah proses *encoder* dimana masukan teks tadi di olah menjadi barisan *bit* data, bukan berupa *byte* atau karakter lagi. Barisan *bit* ini di representasikan dalam *array*.



Gambar 3.13 Desain tampilan *encoder* menggunakan *protocol AX.25*

Proses *encoder* seperti Gambar 3.13 menggunakan *Labview Mathscripts* dan *script* berbasis teks menggunakan mesin *Mathscript RT Module*. *Mathscript Node* digunakan untuk mengevaluasi *script* yang dibuat di Jendela *Labview Mathscript*. Didalam *Mathscript node* yang dibuat terdapat *script* main program dari proses *encoder* atau *bit processing*. Memasukkan program disini menggunakan fungsi *import* agar terintegrasi dengan fungsi lain diluar *main programe*, hal ini berhubungan dengan kinerja dari *Labview* agar tidak terlalu berat dan lama operasinya. Program keseluruhan dari penyusunan *field* untuk menjadi suatu *frame* antara lain:

## 1) Menambahkan *Header*

Menggeser 1 bit ke kiri untuk memungkinkan bit ekstensi HDLC.

### a) Membuat *FLAG*

*FLAG* disini berfungsi sebagai sinkronisasi antar *Frame* dimana *FLAG* dengan menyisipkan angka 1 sebanyak 6 kali diikuti dengan angka 0 diakhir atau dalam hexa direpresentasikan dengan 0x7E dalam biner "01111110".

Perintah dalam program :

```
%create 0b01111110 (FLAG) in byte size
```

```
for i=1:1:64
```

```
    Buffer(size_)=0;
```

```
    %hex2dec('00');
```

```
    size_ = size_ + 1;
```

```
    for j = 1:1:6
```

```
        Buffer(size_)=1;
```

```
        %hex2dec('01');
```

```
        size_ = size_ + 1;
```

```
    end
```

```
    Buffer(size_)=0;
```

```
    %hex2dec('00');
```

```
    size_ = size_ + 1;
```

```
end
```

```
max= s-1;
```

```
%for (int i=0; i < s ; i++)
```

```
Buffer[_size++] = BitSequenceStuffed[i];
```

```
for i = 1:1:max;
```

```
    Buffer(size_) = BitSequenceStuffed(i);
```

```
    size_ = size_+1;
```

```
end
```



b) Menambah *destination callsign* (*callsign* tujuan)

dimana didalamnya terdapat SSID tujuan. Berikut program sebagai perintah dalam menetapkan *callsign* tujuan.

Perintah dalam program :

```
STR_TO_CALLSIGN='YG3GZ';
```

```
c =1;
```

```
Index =1;
```

```
for i=1:1:length(DestCallsign)
```

```
%Append SSID Destination
```

```
bitsequence(Index)= bitshift(unicode2native(DestCallsign(i)),1);
```

```
c = c+1;
```

```
Index = Index+1;
```

```
end
```

```
bitsequence(Index)= hex2dec('60');%ssid_destination;
```

```
Index = Index+1;
```

Dapat dilihat bahwa *callsign* tujuan yang ditetapkan adalah “YG3GZ”.

c) Menambah *source callsign* di indeks setelah *destination callsign*

dimana di dalamnya terdapat SSID number.

Perintah dalam program :

```
STR_FROM_CALLSIGN='YG3EGY';
```

```
%Append Source Callsign
```

```
c =1;
```

```
for i=1:1:length(SrcCallsign) bitsequence(Index)=
```

```
bitshift(uint16(SrcCallsign(i)),1);
```

```
%fprintf(num2str(bitsequence(Index)));
```

```
%fprintf(' ');
```

```
Index = Index+1;
```

```
c =c+1;
```

```
end
```

```
bitsequence(Index)=hex2dec('61');%ssid_source;
```

```
Index = Index+1;
```

Dapat dilihat bahwa *callsign* sumber yang ditetapkan adalah “YG3EGY”.

- d) Menambahkan *control bits* setelah source *callsign* dengan nilai dari control bits adalah 0×03 yang artinya menggunakan tipe *Unnumbered Information Frames*.

Perintah dalam program :

```
CONTROL = '03';%'0x03';
```

```
%Append Control bits
```

```
bitsequence(Index) = hex2dec(CONTROL);
```

```
Index = Index+1;
```

- e) Menambahkan Protokol Identifier (PID) dengan nilai 0×F0 yang berarti tidak menggunakan protokol layer 3 dan ditambahkan setelah Control bits dengan perintah.

Perintah dalam program :

```
PID ='F0';
```

```
%'0xF0';
```

```
%Append Protokol Identifier bitsequence(Index) = hex2dec(PID);
```

```
Index = Index+1;
```

## 2) Menambahkan Pesan/Informasi

Menentukan informasi yang ingin disisipkan ke dalam *Field* informasi. Didalamnya terdapat *convert bitsequence* dimana merupakan proses dari MSB menjadi LSB, dengan CRC .

Perintah dalam program :

```
%Add Message
```

```
c =1;
```

```

for i=1:1: length(txtToSending)
bitsequence(Index) = txtToSending(1,c);
    Index = Index+1;
        c = c+1;
    end
for i=1:1:Index-1
v = (bitsequence(i));
bitsequence(i) = MSB_LSB_swap_8bit(v);
    end

```

- 3) Menambahkan FCS (*Frame Check Sequence*) dengan ditambahkan pada MSB (*Most Significant Bit*)

Perintah dalam program :

```

FCS=typecast(int16(13115),'uint16');
%bin2dec('1100110 0111011');
%13115;
%CRC_CCITT(bitsequence,Index-1);
fprintf('FCS:');
fprintf(num2str(FCS));
fprintf('\n');

```

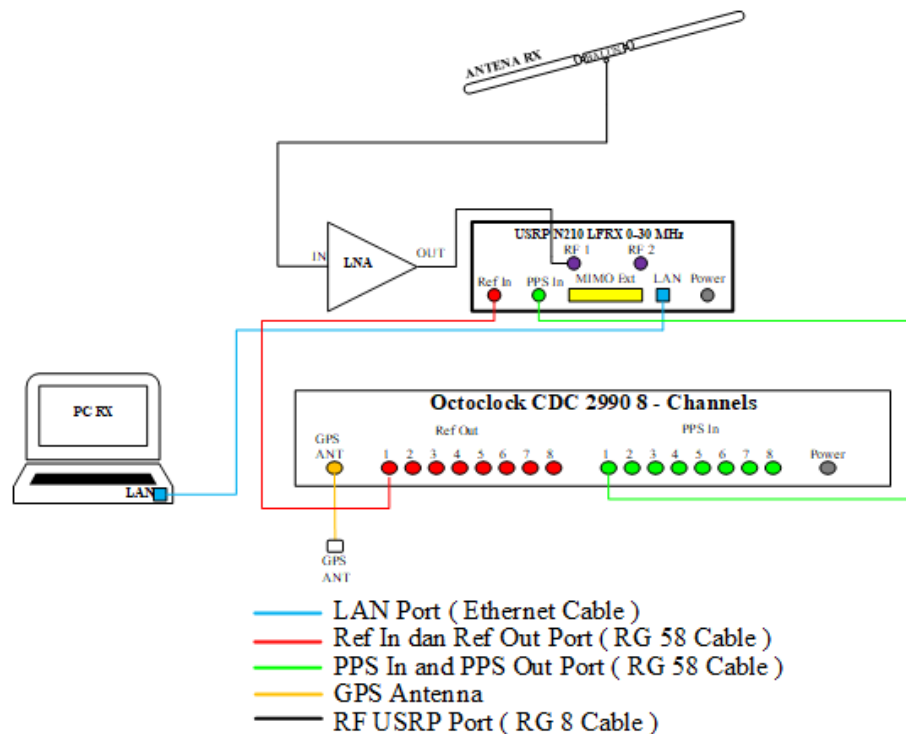
- 4) Hasil Akhir

Hasil dari proses *encoder* adalah *bit sequence* atau barisan bit akhir dan pada saat ini terdapat pembuatan *Flag* dengan biner 01111110 lalu di *recreate* atau dibuat ulang dalam ukuran *byte* . Setelah dibuat maka *flag* di sisipkan. Setelah proses penyusunan menjadi satu *frame* , jadilah barisan bit akhir atau “*final sequence*”.

### 3.4.2 Desain Sistem Penerima

Pada desain sistem penerima berfungsi untuk menerima data teks dari pemancar yang bekerja pada frekuensi 7 MHz. Gambar 3.14 merupakan desain penerima yang digunakan pada sistem ini. Pada desain penerima

terdapat perangkat USRP dengan *software labview* yang digunakan untuk desain, implementasi, pengukuran dan pengambilan data serta laptop sebagai perangkat pengendali sekaligus perangkat yang digunakan untuk menjalankan sistem pada *software labview*. Perangkat *Octoclock* berfungsi untuk mengatur sinkronisasi. Pada perangkat *Octoclock* terdapat antenna *Global Positioning System* (GPS) yang pada waktu pengoperasian harus diletakkan di luar ruangan untuk mendapatkan sinyal GPS yang baik. Perangkat *Low Noise Amplifier* (LNA) berfungsi untuk penguat sinyal pada penerima dari pemancar antenna.

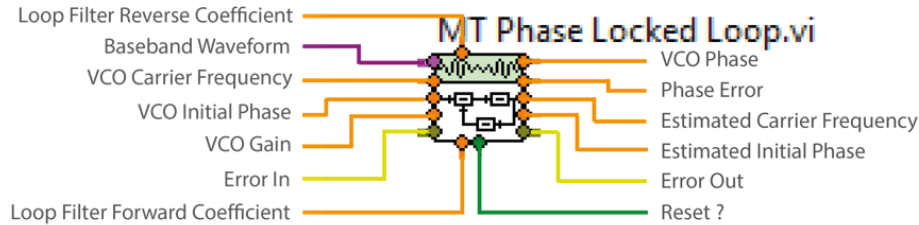


Gambar 3.14 Desain Penerima (RX)

### 3.4.2.1 Desain Sistem Komunikasi Digital HF Penggunaan Tanpa *Protocol AX.25* di Penerima Dengan *Software Labview*

Penelitian pada sistem penerima menggunakan sinkronisasi gelombang pembawa dan AGC yang memiliki fungsi untuk mengontrol secara otomatis penguatan pada tingkat *Intermediate Frequency* (IF) sehingga mendapatkan tingkatan sinyal yang relatif tetap pada keluaran demodulator pada sisi penerima. Sinkronisasi gelombang pembawa pada

software Labview menggunakan modul yang terdapat pada gambar 3.15 sebagai berikut.



Gambar 3.15 Modul MT *Phase-Locked Loop*

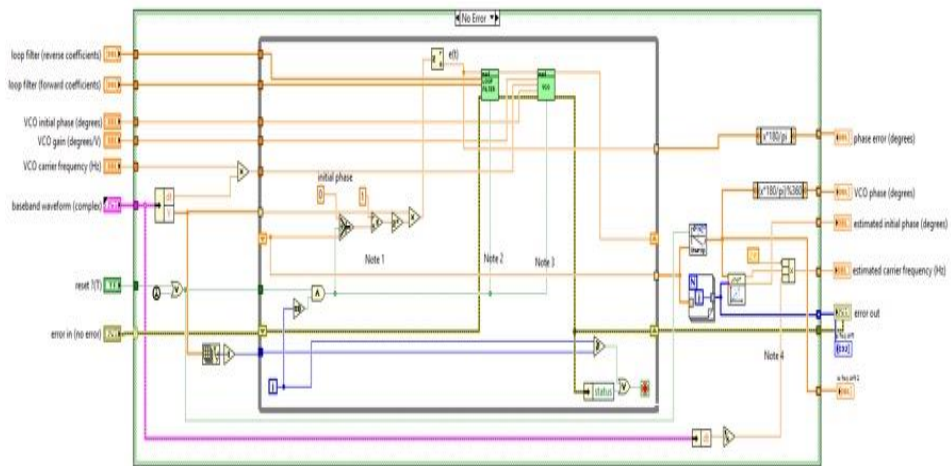
Modul dasar PLL dari LabVIEW ini sudah dirancang dengan fungsi yang sesuai blok diagram secara teoritis. Modul ini memiliki beberapa parameter sebagai masukan atau keluaran untuk sistem, berikut pengertian dasarnya:

- a) **Baseband Waveform:** Menerima sinyal pembawa yang diterima dengan frekuensi dan fase sebagai *estimation* untuk diolah oleh PLL.
- b) **VCO Carrier Frequency:** Menentukan referensi frekuensi pembawa yang digunakan oleh VCO untuk melacak fase dari sinyal pembawa yang diterima, standarnya adalah 0 Hz.
- c) **VCO Initial Phase:** Menentukan referensi awal fase pembawa yang digunakan oleh VCO untuk melacak fase sinyal pembawa yang diterima, standarnya adalah 0 *degree*.
- d) **VCO Gain:** Sebagai penguat yang diterapkan pada *error signal* di dalam VCO sebelum menghasilkan parameter *estimated initial phase*.
- e) **Loop Filter (Reverse Coefficient):** Menentukan *reverse coefficient* dari IIR *Loop Filter*, standarnya adalah (1, -1).
- f) **Loop Filter (Forward Coefficient):** Menentukan *forward coefficient* dari IIR *Loop Filter*, standarnya adalah (1.05, -0.95).
- g) **VCO Phase:** Pengembalian fase secara keseluruhan dari sinyal gelombang pembawa yang diterima dan diestimasi oleh PLL, termasuk efek dari setiap fase dan frekuensi *offsets*.
- h) **Phase Error:** Mengembalikan kesalahan sudut pada sinyal pembawa. *Phase Error* menunjukkan perbedaan fasa antara *input signal* dengan

fasa yang dihasilkan PLL. Parameter ini dihasilkan oleh blok *Phase Detector* (PD).

- i) **Estimated Carrier Frequency:** Perkiraan sinyal yang dihasilkan oleh sistem PLL.
- j) **Estimated Initial Phase:** Perkiraan fasa yang dihasilkan oleh sistem PLL.

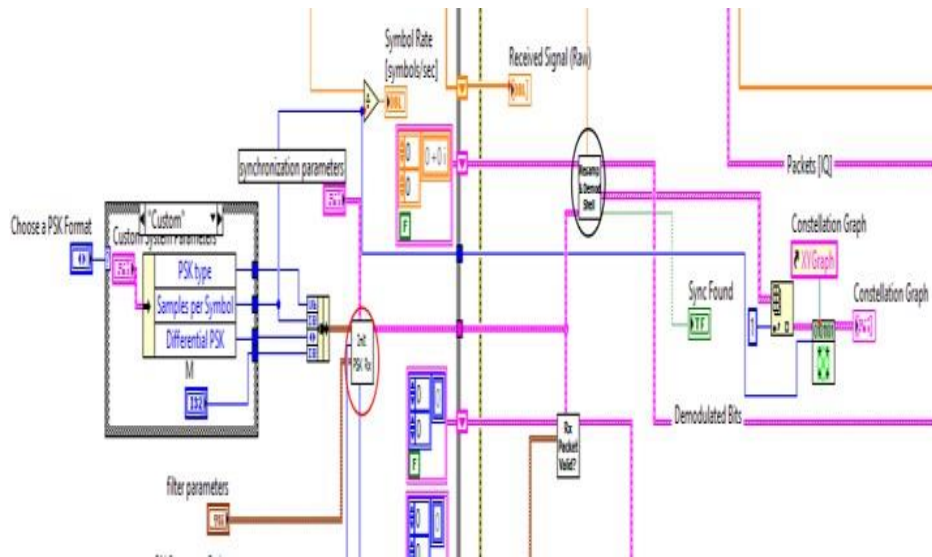
Parameter kontrol utama dalam sistem ini adalah *VCO Carrier Frequency*. Konfigurasi parameter ini, berfungsi menghasilkan replika frekuensi untuk sistem PLL. Sehingga, saat perangkat SDR menerima pergeseran frekuensi dan sudut fasa, PLL akan menjalankan tugasnya dengan melakukan sinkronisasi sinyal replika dari VCO dengan sinyal gelombang pembawa yang diterima. Indikator yang dihasilkan dari modul PLL ini adalah *VCO Phase* dan *Phase Error*. Kedua indikator yang dihasilkan dari sistem tersebut telah dianalisa pada penelitian sinkronisasi gelombang untuk sistem komunikasi digital HF. Blok diagram sistem dari modul PLL dapat dilihat pada gambar 3.16 sebagai berikut.



Gambar 3.16 Desain Blok Diagram Modul PLL

Pada desain gambar tersebut, menunjukkan bahwa masukan sinyal *baseband* yang diterima, akan diteruskan menuju blok *Loop Filter* (LF) setelah melalui bagian *Phase Detector* (PD), yaitu jalur sebelum melewati  $e(t)$ . Kemudian, setelah dari blok LF akan diteruskan ke blok VCO untuk menghasilkan yang diarahkan kembali menuju bagian PD. Proses ini akan

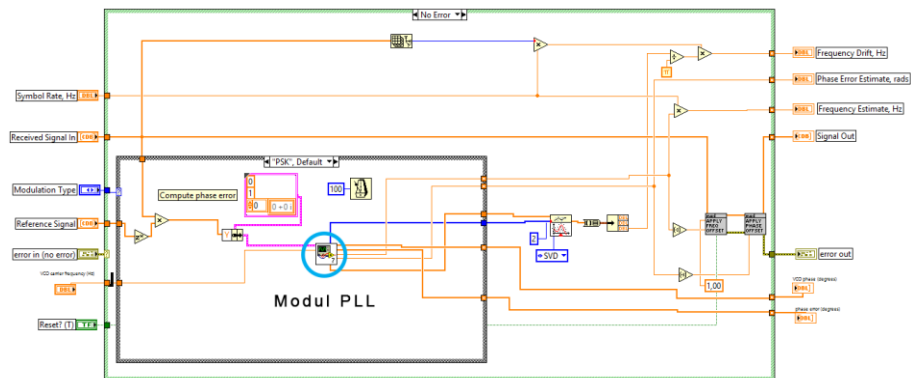
diulang terus sampai bagian PD tidak mendeteksi adanya perbedaan fase antar keluaran VCO dengan sinyal yang diterima, yang mana fase ini dinamakan dengan *steady state*.



Gambar 3.17 Desain Sistem Tanpa *Protocol* di Sisi Penerima

Gambar 3.17 menjelaskan proses demodulasi. Dimana sinyal yang diterima sebelumnya masih dalam sinyal yang berbentuk sinyal informasi *bandpass*. Untuk dapat merubah sinyal informasi menjadi sinyal *baseband* diperlukan resample demodulasi sinyal.

Konsep PLL tidak lepas dari konfigurasi sisi penerima pada sebuah sistem komunikasi. Sehingga, modul PLL ini telah diintegrasikan dengan *demodulator* untuk komunikasi digital pada perangkat lunak *Labview*. Modul PLL diintegrasikan pada bagian *baseband* sistem demodulasi, tepatnya di bagian sub modul *MT Demodulation*. Modul *MT Demodulation* ini memproses data gelombang kompleks seperti *matched filter*, dan *error correction* pada frekuensi dan fasa, dapat digambarkan pada gambar 3.18 sebagai berikut.



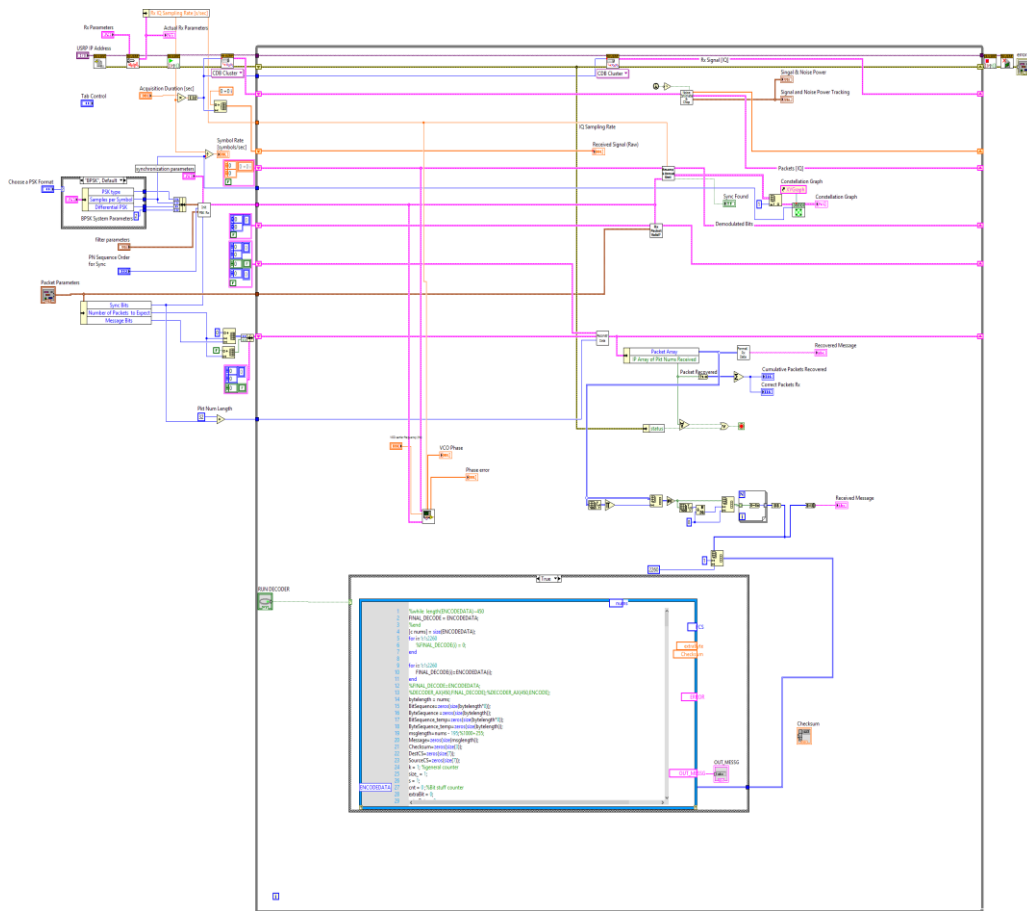
Gambar 3.18 Integrasi Desain PLL dengan Sistem Tanpa *Protocol* di Sisi Penerima

Pada tampilan integrasi desain dengan sistem tersebut, modul PLL diintegrasikan pada sub *frequency and phase offset* yang telah dimodifikasi agar tidak terjadi tumpang tindih antar sistem karena adanya komponen atau modul yang kedua sistem tersebut miliki. Modul PLL yang digunakan dalam penelitian sinkronisasi gelombang pembawa menggunakan perangkat lunak LabVIEW ini menghasilkan 2 (dua) parameter yang dapat dianalisa pada penelitian ini, yaitu *VCO Phase* dan *Phase Error*.

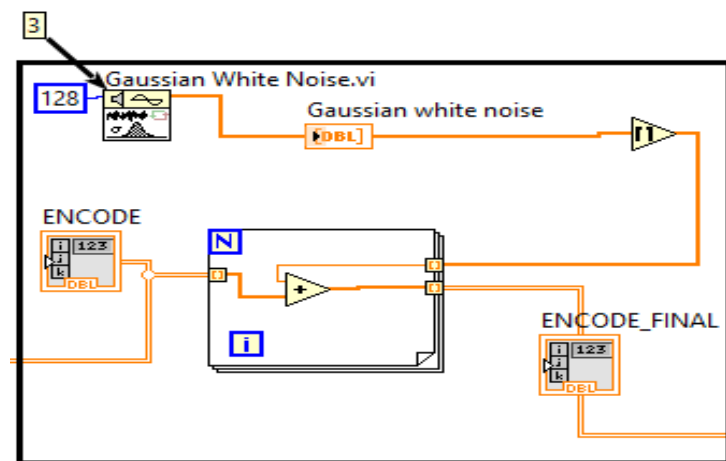
### 3.4.2.2 Desain Sistem Komunikasi Digital HF Penggunaan Dengan *Protocol AX.25* di Penerima Dengan *Software Labview*

Sistem komunikasi digital HF pada *software labview* di penerima terdapat pada gambar 3.19 dengan desain menggunakan *protocol AX.25*. Sebelum ke proses *decoder* terdapat proses penambahan noise generator, dimana seolah-olah terjadi adanya *noise* pada saat proses transmisi dari *encoder* ke *decoder*, hal ini dirancang sesuai dengan sebagaimana sistem komunikasi secara nyata terjadi yaitu adanya *noise* apabila melalui media berupa udara. Hasil dari proses *encoder* adalah berupa bit *array* yang ditambah dengan nilai dari *noise* generator yang didistribusikan. Generator *noise* disini menggunakan *Gaussian white noise* dengan 128 *sample* menyesuaikan *default* dari mode *Labview* sendiri. Setelah ditambahkan maka dapat menghasilkan *array encode final*. Mengenai desain *noise* terdapat pada gambar 3.20.





Gambar 3.19 Desain Sistem Dengan *Protocol* di Sisi Penerima



Gambar 3.20 Desain Noise Pada *Software Labview*

Proses *decoder* dimana *input* yang masuk merupakan barisan bit (*bitsequence*) hasil dari *encoder* yang telah ditambahkan dengan *noise* seperti Gambar 3.20. Lalu proses berjalan pada *Mathscript node* yang dibuat dan didalamnya terdapat *script* main program dari proses *decoder*. Terdapat

beberapa tahapan dalam proses *decoder* sehingga barisan bit menjadi *string*, secara umum yaitu proses ini kebalikan dari proses *encoder*:

#### 1) Menemukan dan Menghilangkan *Flag*

Proses ini merupakan awal pada saat *decoder* menerima barisan bit akhir dari pengirim lalu terjadi sinkronisasi antara *frame* satu dengan *frame* yang lain.

Perintah dalam program :

```
for i =1:1:k-1
if ByteSequence(i) ~= bin2dec('01111110') pastFlag = true;
ByteSequence_temp(cnt) = ByteSequence(i);
cnt = cnt +1;
elseif pastFlag == true break;
end
end
```

#### 2) FCS (*Frame Check Sequence*)

Melakukan kalkulasi CRC untuk menentukan FCS apakah sesuai dengan FCS dipengirim atau tidak.

Perintah dalam program :

```
FCS =
13115;%typecast(int16(13115),'uint16');
%CRC_CCITT(ByteSequence, k-2);
Checksum(1) = ByteSequence(k-2);
Checksum(2) = ByteSequence(k-1);
if Checksum(1) ~=bitand(bitshift(FCS,- 8),hex2dec('FF'))
fprintf('Error in Checksum 1 : ');
fprintf('\n');
end
ERROR = 'tidak';
if Checksum(2) ~= (bitand(FCS,hex2dec('FF'))) fprintf('Error in
Checksum 2 : ');
ERROR='ya';
```

```

fprintf('\n');
else
fprintf('Correct checksum 2\n ');
end

```

### 3) LSB to MSB

Perintah dalam program :

```

for i=1: i : bytelength ByteSequence_temp(i)= 0;
end
%for (int i=0; i < k-2 ; i++) ByteSequence_temp[i]=
MSB_LSB_swap_8bit(ByteSequence[i]);
for i=1: 1 : k-3
ByteSequence_temp(i)=
MSB_LSB_swap_8bit(ByteSequence(i));
fprintf(num2str(ByteSequence_temp(i));
fprintf(' ');
end

```

### 4) Recover Header

Setelah merubah LSB ke MSB maka dilanjutkan ke pemulihan komponen dari *header*.

Perintah dalam program :

```

for i =1:1:6
DestCS(i)= bitshift(ByteSequence_temp(cnt),-1);
cnt =cnt+1;
end
%Append Source Callsign
%for(int i=0; i < 6; i++)
SourceCS[i]=char(ByteSequence_temp[cnt++]>>1);
for i=1:1:6
SourceCS(i)=bitshift(ByteSequence_temp(cnt),- 1);

```

```

cnt =cnt+1;
end
%Append SSID Source cnt =cnt +1;
%Append Control bits cnt =cnt +1;
%Append Protocol Identifier cnt =cnt +1;
%Recover message s = k-2-cnt;
fprintf('Final decoded Message\n');

```

#### 5) Menambahkan *SSID* Sumber

Menambahkan SSID sumber ini diambil dari *callsign* sumber yang diberikan pada saat proses pada transmitter.

#### 6) Menambahkan *control bits*

Menambahkan bit control disini disesuaikan nilainya seperti apa yang sudah di sesuaikan di pemancar / transmitter.

#### 7) Menambahkan *Protocol Identifier*

Menambahkan PID disini disesuaikan nilainya seperti apa yang sudah di sesuaikan di pemancar / transmitter.

#### 8) *Recover Message*

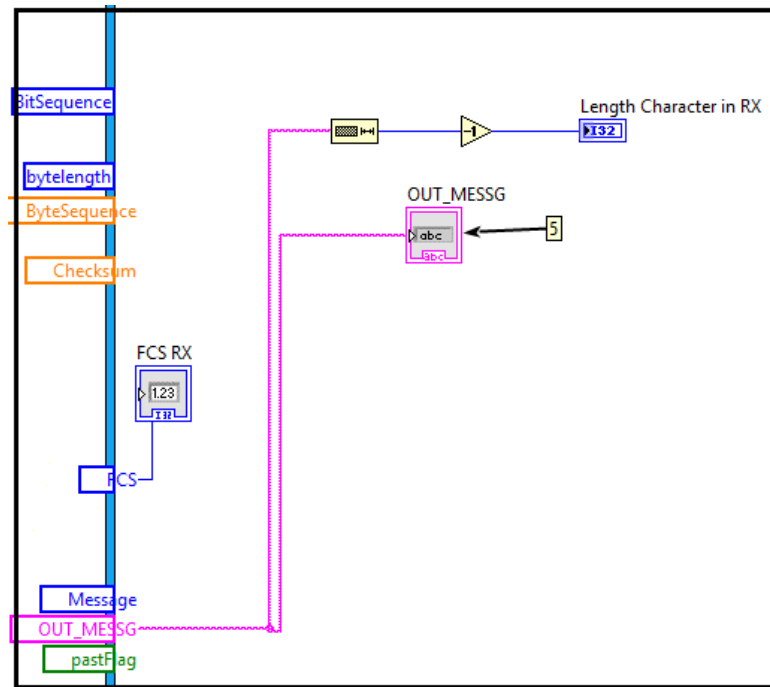
```

1  %while length(ENCODEDATA)~4;
2  FINAL_DECODE = ENCODEDATA;
3  %end
4
5  %FINAL_DECODE=ENCODEDATA;
6  %DECODER_AX(450,FINAL_DECOI
7  bytelength = 450+256+256+2
8  BitSequence=zeros(size(bytelengt
9  ByteSequence =zeros(size(bytelen
10 BitSequence_temp=zeros(size(bytr
11 ByteSequence_temp=zeros(size(by
12 Message=[]zeros(size(256));
13 Checksum=zeros(size(256));

```

Gambar 3.21 Mathscript node pada *labview*

Proses berikutnya merupakan *output* hasil akhir dari semua proses, dimana *Indicator* OUT\_MSSG bernilai *string* dan muncul pada panel / *user interface* yang berisi karakter yang asli atau yang dikirim.



Gambar 3.22 Desain Output

### 3.5 Desain Parameter

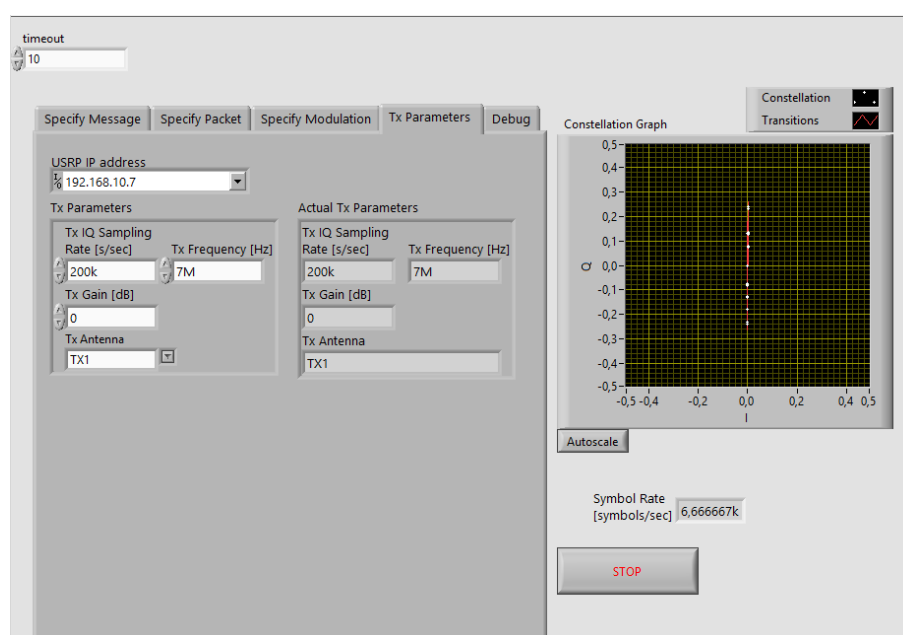
Indonesia adalah salah satu Negara Asia Pasifik yang berlokasi di wilayah 3 yang memanfaatkan pita 7,0 – 7,2 MHz untuk radio amatir<sup>[25]</sup>. Bandwidth maksimum yang diizinkan untuk komunikasi data melalui kanal HF adalah 15 kHz<sup>[26]</sup>. Operator radio amatir IAR menggunakan daya pancar maksimum yang diizinkan pada frekuensi dibawah 30 MHz menjadi 100 Watt<sup>[27]</sup>. Signal to Noise Ratio (SNR) minimum yang diizinkan dalam sistem komunikasi adalah 10 dB<sup>[28]</sup>.

#### 3.5.1 Desain Parameter di Pemancar

Desain parameter sistem komunikasi digital HF pada sisi pemancar dapat di sampaikan pada tabel 3.2. dan tampilan input parameter di pemancar pada *software labview* dapat di tampilkan pada gambar 3.23 sebagai berikut.

Tabel 3.2 Desain Parameter Sistem Komunikasi Digital HF pada sisi Pemancar

No	Parameter	Nilai
1	Frekuensi Kerja	7 MHz
2	Bandwidth	10 KHz
3	IQ Rate	200 Ksymbol/sec
4	Modulasi	BPSK
5	Samples per Symbol	30
6	Filter	Root Raised Cosine
7	Roll of Factor	0.5
8	Power Pemancar	3 Watt
9	Guard Bits	30
10	Sync Bits	20
11	Message Bits	128



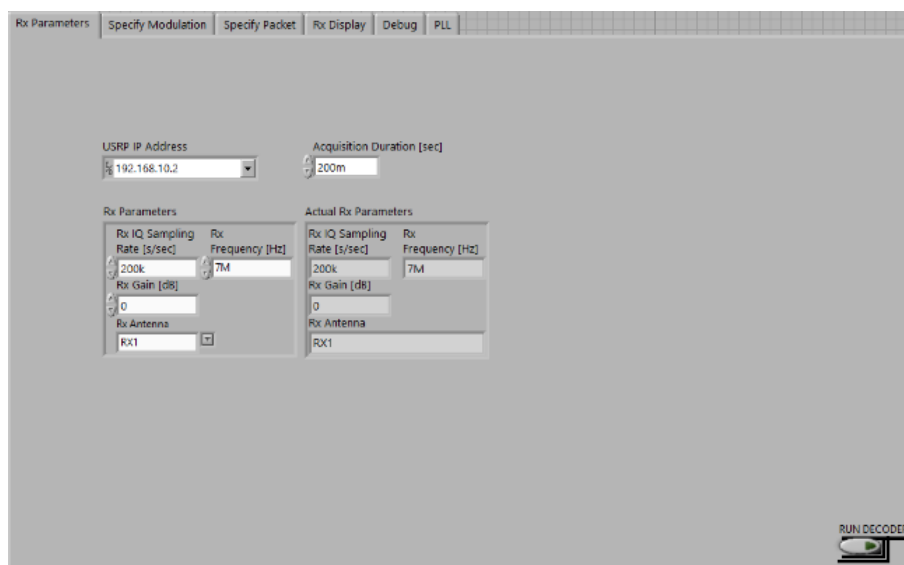
Gambar 3.23 Tampilan Input Parameter di Pemancar Pada *Software Labview*

### 3.5.2 Desain Parameter di Penerima

Desain parameter sistem komunikasi digital HF pada sisi penerima dapat di sampaikan pada tabel 3.3. dan tampilan input parameter di penerima pada *software labview* dapat di tampilkan pada gambar 3.24 sebagai berikut.

Tabel 3.3 Desain Parameter Sistem Komunikasi Digital HF pada sisi Pemancar

No	Parameter	Nilai
1	Frekuensi Kerja	7 MHz
2	Bandwidth	10 KHz
3	IQ Rate	200 Ksymbol/sec
4	Modulasi	BPSK
5	Samples per Symbol	30
6	Filter	Root Raised Cosine
7	Roll of Factor	0.5
8	Power Pemancar	3 Watt
9	Guard Bits	30
10	Sync Bits	20
11	Message Bits	128



Gambar 3.24 Tampilan Input Parameter di Penerima Pada *Software Labview*

### 3.6 Desain *Link Budget*

Desain *Link budget* diperlukan untuk mengukur berapa daya yang dibutuhkan agar dapat melakukan transmisi dengan baik. Pada desain *link budget* untuk jarak jauh kali ini di desain untuk link Surabaya-Malang

dengan penggunaan frekuensi kerja 7 MHz. Berikut untuk rugi – rugi propagasi *skywave* dapat di tampilkan pada tabel 3.3 sebagai berikut.

Tabel 3.3 Desain perhitungan *skywave* pathloss dengan menggunakan asumsi redaman tinggi

Parameter	Nama Parameter	Nilai	Keterangan
a	Jarak garis lurus <i>Line Of Sight</i> (LOS)	0,05 Km	Panjang horizontal
h	Tinggi Ionosfer (Km)	300 Km	Diasumsikan Ketinggian Virtual Ke Lapisan F
D	Jarak Miring (Km)	300 Km	
d	Jarak Lintasan (Km)	600 Km	Jarak Tempuh Gelombang
f	Frekuensi (MHz)	7 MHz	
$L_a$	<i>Absorption loss</i>	20 dB	Bernilai sekitar 0-20 dB ( $L_a \propto 1/f^2$ )
$L_b$	FSL ( <i>Free Space Loss</i> )	104,90 dB	$L_b = 20\log\left(\frac{4\pi d}{\lambda}\right)$
$L_g$	<i>Multihop ground reflection loss</i>	0 dB	Bernilai antara 1-3dB/ reflection (Asumsi tidak ada reflection)
$L_p$	<i>Polarization coupling loss</i>	6 dB	Bernilai antara 3-6 dB 6dB digunakan untuk asumsi tertinggi
$L_q$	<i>sporadic E obstruction loss</i>	1 dB	<i>Function of sporadice critical frequency;</i> Berniai antara 0-1 dB
$G_f$	<i>Horizon Focus Gain</i>	0 dB	Bernilai antara 0-1 dB (Asumsi tidak terjadi pengutan horizon)

Dari desain perhitungan di atas didapat bahwa rugi-rugi total dari propagasi *skywave* mencapai 131.90 dB. Perhitungan di atas tersebut adalah perhitungan dengan menggunakan asumsi terburuk. Untuk



*Absorption Loss* digunakan 20 dB, Sporadic E Obstruction Loss diberlakukan 1 dB untuk asumsi tertingginya.

Dari perhitungan tersebut di atas, kemudian dilanjutkan untuk desain perhitungan *Link Budget* yang dibutuhkan. Untuk parameter dan perhitungannya dapat dilihat pada Tabel 3.4 sebagai berikut.

Tabel 3.4 Desain Perhitungan *Link Budget*

Parameter	Nama Parameter	Nilai	Keterangan
$P_{TX}$	Daya Pancar	34.77 dBm	Daya keluaran dari High Power Amplifier
$L_{FTX}$	Redaman kabel pemancar	0.19 dB	1.6 dB / 100 meter Panjang kabel = 12 meter (RG-8)
$L_{FRX}$	Redaman kabel penerima	0.9 dB	3 dB / 100 meter Panjang kabel = 30 meter (RG-58)
$G_{ATX}$	Gain pemancar	5.3 dB	Perhitungan Proplab
$G_{ARX}$	Gain penerima	5.3 dB	Perhitungan Proplab
$G_{RX}$	Gain LNA	20 dB	Dengan Catu daya 12 V
$A_T$	Total Pathloss	131.90 dB	<i>Skywave Pathloss</i>
EIRP	<i>Effective Isotropic Radiated Power</i>	39.88 dBm	$EIRP = P_{TX} - L_{FTX} + G_{ARX}$
$P_{RX}$	Daya Penerima	-92.02 dBm	$P_{RX} = EIRP - A_T$
F	Noise Figure	3.1 dB	$F = F1 + ((F2-1)/G1)$
k	Konstanta Blotzman	$1.38 \times 10^{-23}$ J/K	
$T_o$	Noise Temperature	290° K	
B	Bandwidth	10 kHz	
$N_{Out}$	<i>Noise Output</i>	-125.58 dBm	$N_o = F \times G_{ARX} \times k \times T_o \times B$
SNR	<i>Signal To Noise Ratio</i>	39.05 dB	$SNR = P_{TX} + G_{ATX} + G_{ARX} - N_o - A_T$
$SNR_{min}$	Minimum SNR	10 dB	

Tabel 3.4 Desain Perhitungan *Link Budget* (lanjutan)

Parameter	Nama Parameter	Nilai	Keterangan
Dynamic Range		34.67 dB	DR = SNR - SNR <sub>min</sub>
RX Sensitivity		-76 dBm	Sensitivitas USRP
Fade Margin		-16.02 dB	Fade Margin = P <sub>RX</sub> - RX Sensitivity

### 3.7 Indikator Kinerja

Untuk menghasilkan penelitian yang baik pada sistem komunikasi digital HF diperlukan adanya kinerja sistem. Kinerja yang diukur pada sistem komunikasi digital HF yaitu penggunaan tanpa *protocol* dan dengan *protocol*. Kinerja sistem ini terdiri dari beberapa indikator kinerja diantaranya sebagai berikut :

#### 3.7.1 *Signal to Noise Ratio* (SNR)

Penghitungan hasil pengukuran dalam menentukan nilai SNR dapat dilakukan menggunakan persamaan sebagai berikut<sup>[6]</sup>.

$$SNR (dB) = 10 \log 10 \frac{S}{N} \quad (3.1)$$

#### 3.7.2 *Bit Error Rate* (BER)

Kesalahan yang terjadi selama proses transmisi data dilakukan dengan cara membandingkan masukan total yang dikirim dengan jumlah bit yang error. BER dapat dihitung menggunakan persamaan sebagai berikut.

$$BER = \frac{\text{Jumlah Bit Error}}{\text{Jumlah Bit Yang Dikirim}} \quad (3.2)$$

#### 3.7.3 *Character Error Rate* (CER)

Sebuah karakter tersusun atas 8 bit dan dapat berupa huruf, angka, tanda baca seperti titik, koma, tanda seru, tanda tanya, tanda kutip, symbol dan spasi. CER dapat dihitung menggunakan persamaan sebagai berikut.

$$CER = \frac{\text{Jumlah Karakter Error}}{\text{Jumlah Karakter Yang Dikirim}} \quad (3.3)$$

#### 3.7.4 *Frame Error Rate (FER)*

Frame berupa paket yang terdiri dari beberapa karakter. Satu Frame dalam Protokol AX.25 terdiri dari 255 bit<sup>[18]</sup>. FER dapat dihitung menggunakan persamaan sebagai berikut.

$$FER = \frac{\text{Jumlah Frame Error}}{\text{Jumlah Frame Yang Dikirim}} \quad (3.4)$$

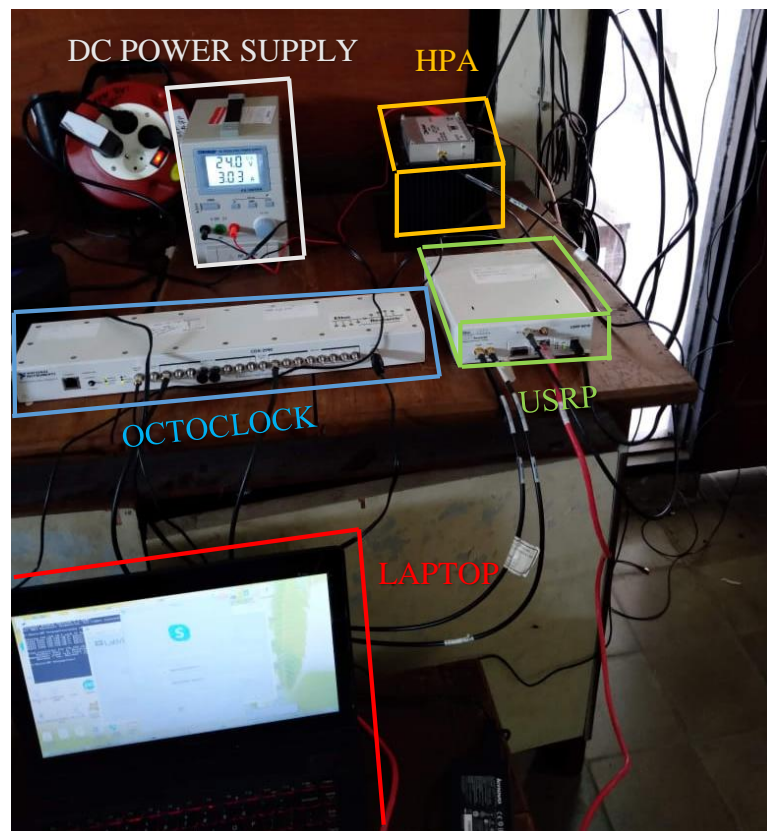
## BAB 4

### HASIL DAN PEMBAHASAN

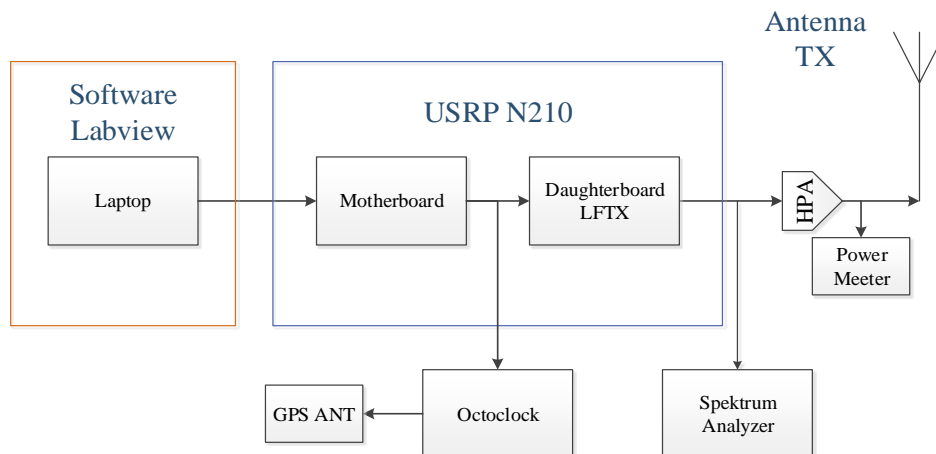
#### 4.1 Hasil Implementasi Sistem Keseluruhan

Hasil implementasi sistem komunikasi digital HF secara keseluruhan pada pemancar dan penerima menggunakan parameter tanpa *protocol AX.25* dan dengan *protocol AX.25* berhasil.

Pemasangan perangkat untuk uji coba sistem komunikasi digital HF dengan implementasi tanpa dan dengan *protocol AX.25* pada sisi pemancar dan penerima adalah sama. Konfigurasi perangkat pemancar dapat ditunjukkan pada gambar 4.1, sedangkan untuk blok diagram dari sistem pengukuran pemancar dapat ditunjukkan pada gambar 4.2. Konfigurasi perangkat penerima dapat ditunjukkan pada gambar 4.3, sedangkan untuk blok diagram dari sistem pengukuran penerima dapat ditunjukkan pada gambar 4.4.



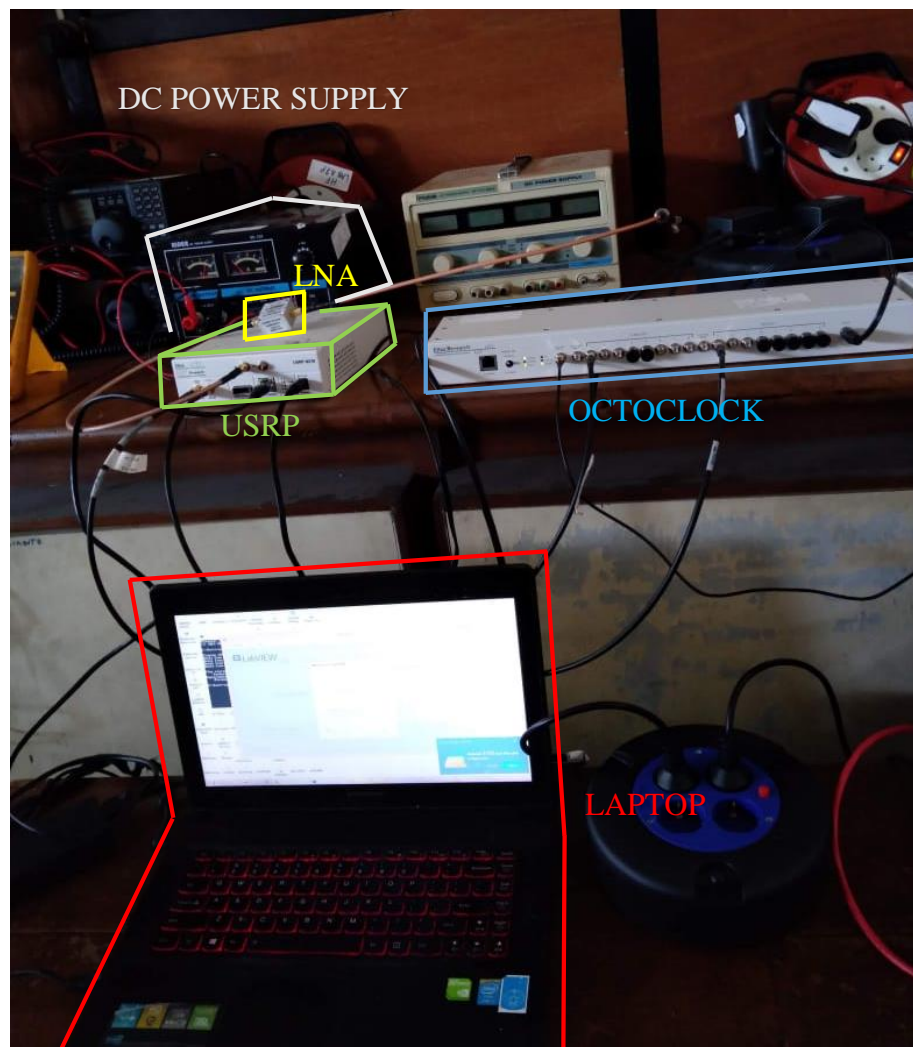
Gambar 4.1 Konfigurasi Perangkat Pemancar



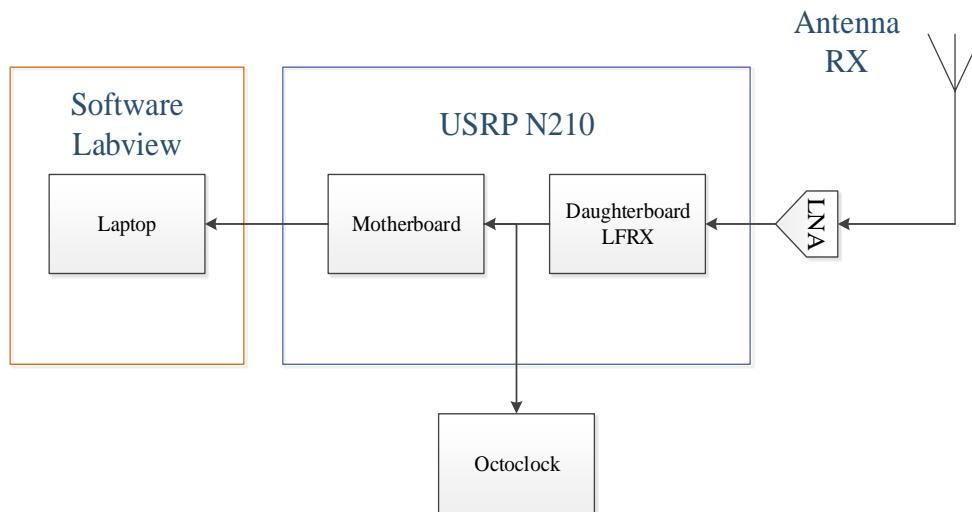
Gambar 4.2 Blok Diagram Sistem Pengukuran di Pemancar

Berdasarkan gambar 4.2 dapat di jelaskan proses sistem pengukuran yang dilakukan pada penelitian ini memiliki beberapa tahapan. Pertama instalasi perangkat keseluruhan terlebih dahulu. Perangkat yang disiapkan ada laptop, USRP, *Daughterboard LFTX*, *Octoclock*, GPS Antena, DC *Power Supply*, HPA, Antena *Dipole  $\lambda/2$* , *port coaxial* dengan kabel RG 8, *port SMA* dengan kabel RG 58, dan kabel *Gigabit Ethernet*. Setelah semua perangkat siap selanjutnya melakukan pengaktifan laptop yang sudah terinstal *software labview* di dalamnya sudah di desain dengan dan tanpa penggunaan protocol AX.25. Kedua siapkan dan nyalakan USRP yang di dalamnya sudah terpasang *daughterboard LFTX* serta sambungkan kabel *gigabit ethernet* dari laptop ke USRP pada *port ethernet*. Setelah tersambung lakukan test konektivitas dengan cara klik *command prompt* dengan tes ping ke IP yang tertera pada USRP. Ketiga siapkan dan nyalakan *octoclock* dengan terpasang antena GPS pada *port GPS ANT* pada *octoclock* serta sambungkan *connector SMA* kabel RG 58 ke USRP. *Port* yang digunakan di USRP yaitu *port Ref In* dihubungkan ke *port Ref Out* pada *Octoclock*, sedangkan *port PPS In* dihubungkan ke *PPS Out* pada *Octoclock*. Keempat perangkat *Octoclock* juga disambungkan ke GPS Antena yang memiliki fungsi sebagai tanda lokasi wilayah saat melakukan pemancaran. Kelima pada port RF 1 USRP di sambungkan ke HPA dengan *connector SMA to PL* dengan kabel RG 58 yang dibangkitkan dengan DC *Power Supply*. Setelah itu *connector* dan kabel disambungkan

ke *Power Meter* untuk melakukan pengukuran daya pancar pada *port TX* di *Power Meter*, sedangkan untuk *port* antena di sambungkan ke antena *Dipole  $\lambda/2$*  dengan *conector coaxial* dengan kabel RG 8. Di *USRP port RF 2* di sambungkan ke *Spectrum Analyzer* untuk mengukur frekuensi sinyal yang di pancarkan sudah sesuai dengan desain dan aturan IAR serta Kemkominfo. Setelah semua tersambung dengan baik, maka pada *software labview* di inputkan data teks yang sudah di desain tanpa dan dengan penggunaan protocol AX.25 sesuai dengan variasi teks yang digunakan. Setelah diinputkan maka siap untuk di pancarkan.



Gambar 4.3 Konfigurasi Perangkat Penerima



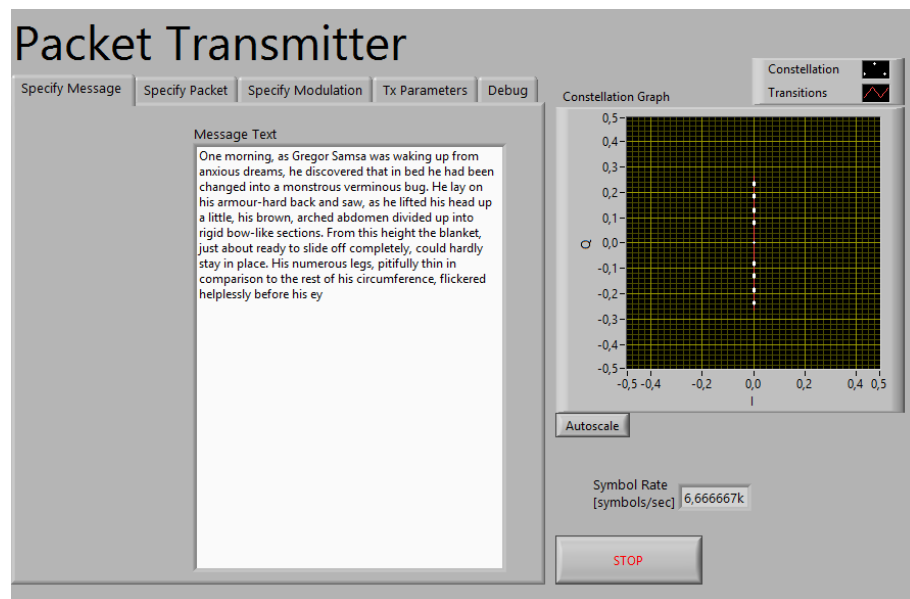
Gambar 4.4 Blok Diagram Sistem Pengukuran di Penerima

Berdasarkan gambar 4.4 dapat di jelaskan proses sistem pengukuran yang dilakukan pada penelitian ini memiliki beberapa tahapan. Pertama instalasi perangkat terlebih dahulu. Perangkat yang disiapkan ada laptop, USRP, *Daughterboard LFRX*, *Octoclock*, GPS Antena, DC Power Supply, LNA, Antena *Dipole  $\lambda/2$* , *port coaxial* dengan kabel RG 8, *port SMA* dengan kabel RG 58, dan kabel *Gigabit Ethernet*. Setelah semua perangkat siap selanjutnya melakukan pengaktifan laptop yang sudah terinstal *software labview* di dalamnya sudah di desain dengan dan tanpa penggunaan protocol AX.25. Kedua siapkan dan nyalakan USRP yang di dalamnya sudah terpasang *daughterboard LFRX* serta sambungkan kabel *gigabit ethernet* dari laptop ke USRP. Setelah tersambung lakukan test konektivitas dengan cara klik *command prompt* dengan tes ping ke IP yang tertera pada USRP. Ketiga siapkan dan nyalakan *octoclock* dengan terpasang antenna GPS pada port GPS ANT pada *octoclock* serta sambungkan *conector SMA* dengan kabel RG 58. *Port* di USRP yang digunakan yaitu *port Ref In* yang dihubungkan ke *port Ref Out* pada *Octoclock*, sedangkan *port PPS In* dihubungkan ke *PPS Out* pada *Octoclock*. Keempat perangkat *Octoclock* juga disambungkan ke GPS Antena yang memiliki fungsi sebagai tanda lokasi wilayah saat melakukan pemancaran. Kelima pada port RF 1 USRP di sambungkan ke LNA menggunakan *conector SMA to PL* dengan kabel RG 58. Setelah semua

tersambung dengan baik, maka pada *software labview* dapat menerima data teks yang sudah di desain tanpa dan dengan penggunaan protocol AX.25 sesuai dengan variasi teks yang digunakan.

#### 4.1.1 Hasil Implementasi Sistem Pemancar Tanpa *Protocol* AX.25

Pada penelitian ini di dapatkan hasil implementasi sistem komunikasi digital HF pada sisi pemancar tanpa penggunaan *protocol* AX.25 berhasil dikirimkan menggunakan *software labview* yang diintegrasikan ke USRP. Pengujian dilakukan dengan cara melakukan pengiriman data teks sebanyak 10 kali pada masing-masing variasi karakter. Variasi karakter yang digunakan yaitu 500, 700, 1000, 1500 dan 2000 karakter.

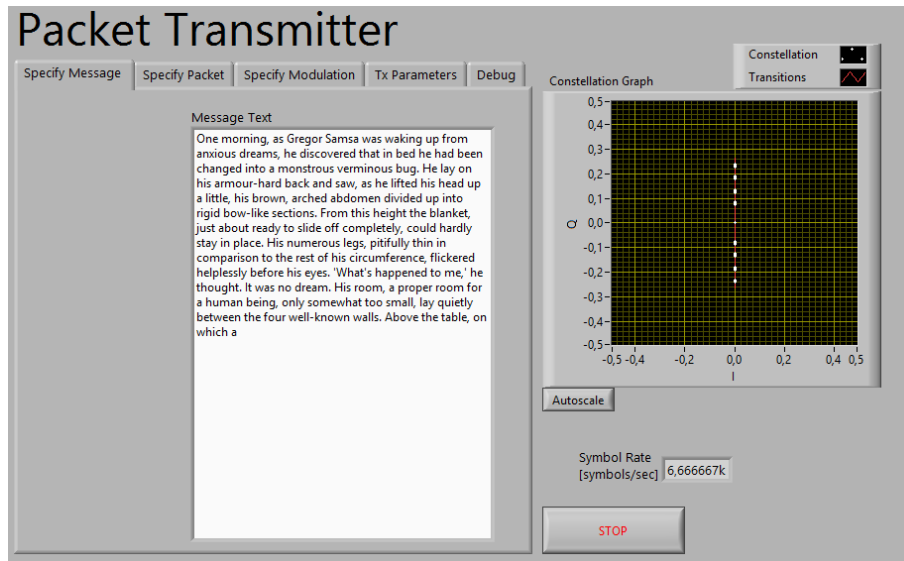


Gambar 4.5 Implementasi Pengiriman Teks 500 Karakter Tanpa *Protocol* Dengan rincian teks sebagai berikut :

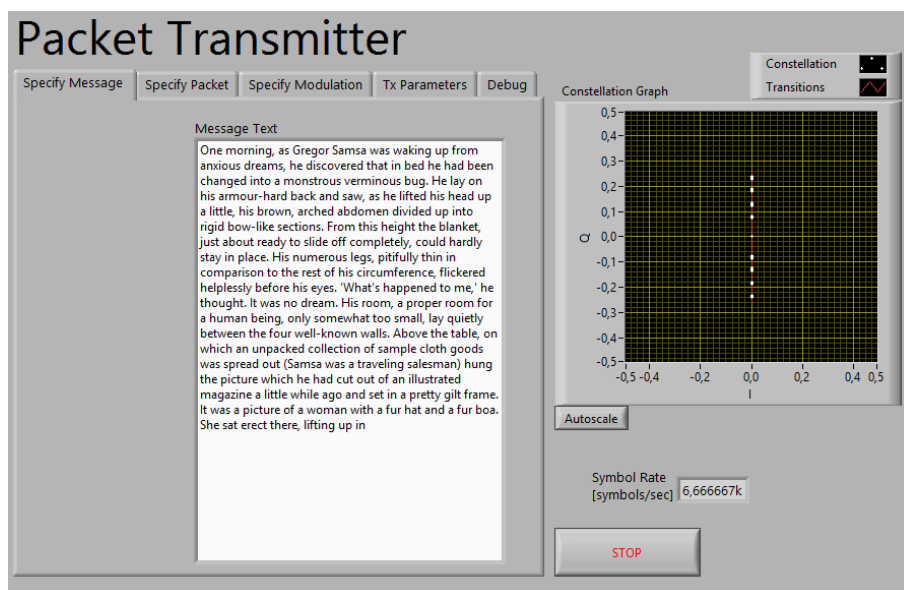
“One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could



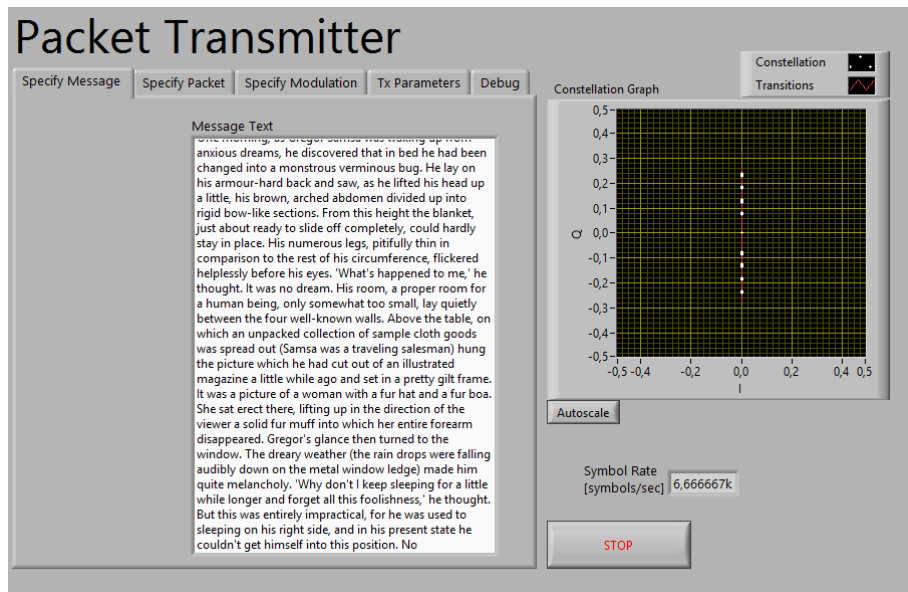
hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his ey”.



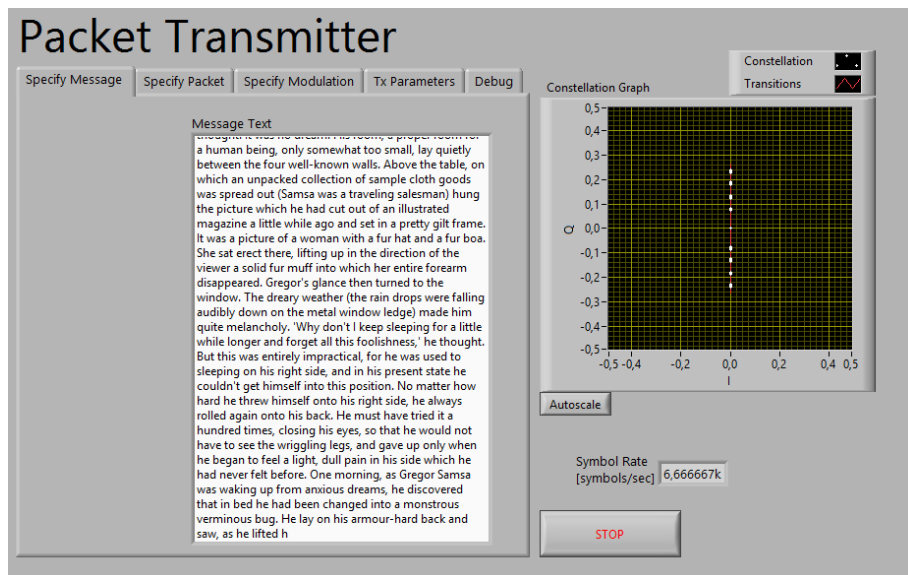
Gambar 4.6 Implementasi Pengiriman Teks 700 Karakter Tanpa *Protocol*



Gambar 4.7 Implementasi Pengiriman Teks 1000 Karakter Tanpa *Protocol*



Gambar 4.8 Implementasi Pengiriman Teks 1500 Karakter Tanpa *Protocol*

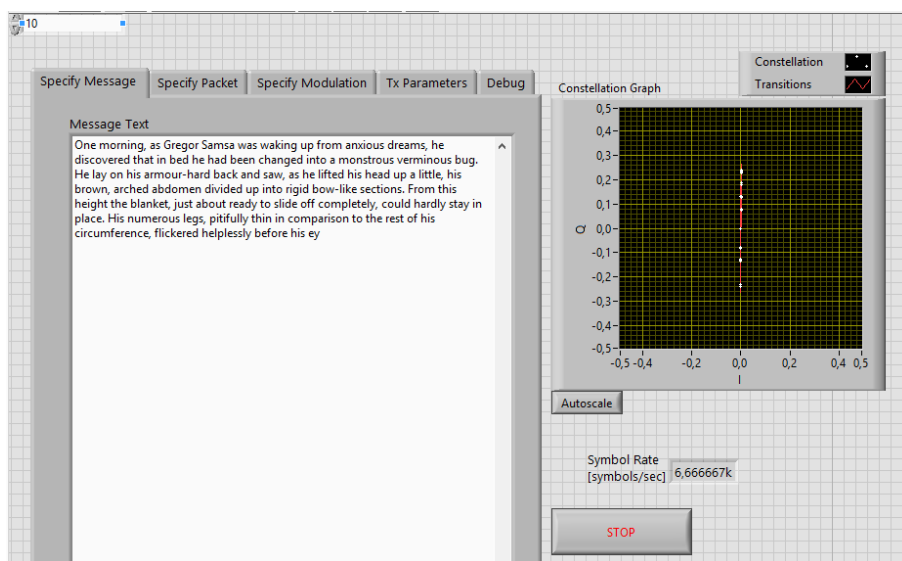


Gambar 4.9 Implementasi Pengiriman Teks 2000 Karakter Tanpa *Protocol*

Gambar 4.5 tampilan hasil pengiriman teks 500 karakter, gambar 4.6 tampilan hasil pengiriman teks 700 karakter, gambar 4.7 tampilan hasil pengiriman teks 1000 karakter, gambar 4.8 tampilan hasil pengiriman teks 1500, dan gambar 4.9 tampilan hasil pengiriman teks 2000 karakter pada sisi pemancar tanpa menggunakan *protocol* AX.25 pada *software labview* dengan parameter yang sudah di jelaskan pada sub bab 3.5.1.

#### 4.1.2 Hasil Implementasi Sistem Pemancar Dengan Protokol AX.25

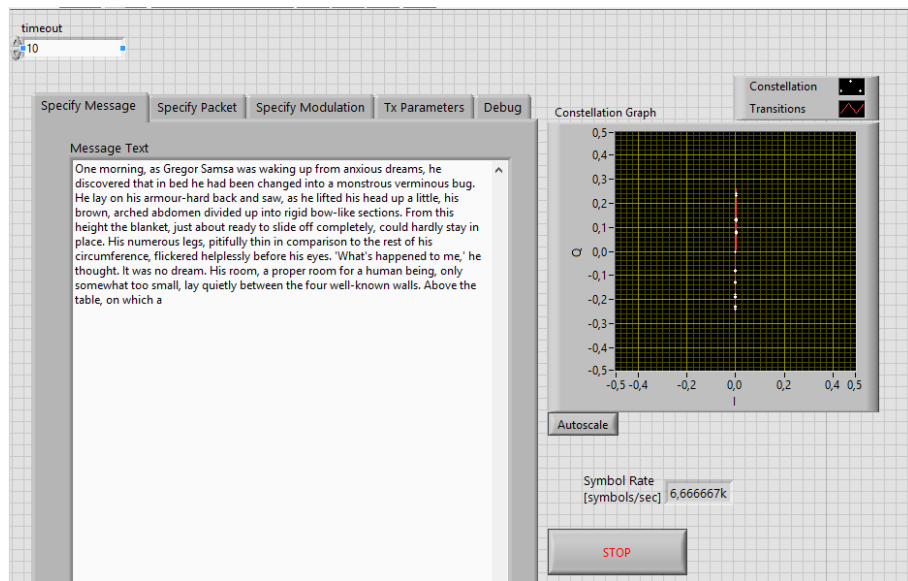
Pada penelitian ini di dapatkan hasil implementasi sistem komunikasi digital HF pada sisi pemancar dengan penggunaan *protocol* AX.25 berhasil dikirimkan menggunakan *software labview* yang diintegrasikan ke USRP. Pengujian dilakukan dengan cara melakukan pengiriman data teks sebanyak 10 kali pada masing-masing variasi karakter. Variasi karakter yang digunakan yaitu 500, 700, 1000, 1500 dan 2000 karakter.



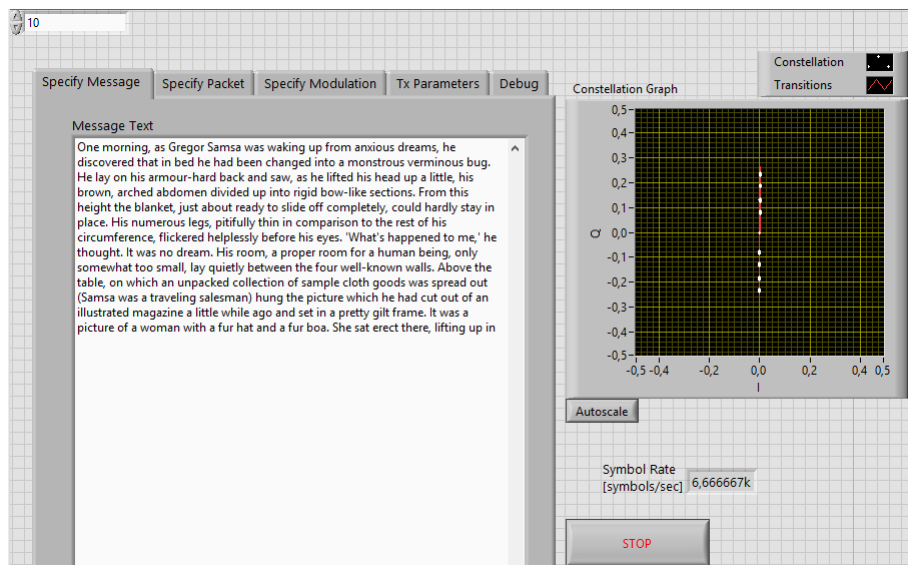
Gambar 4.10 Implementasi Pengiriman Teks 500 Karakter Dengan *Protocol*

Dengan rincian teks sebagai berikut :

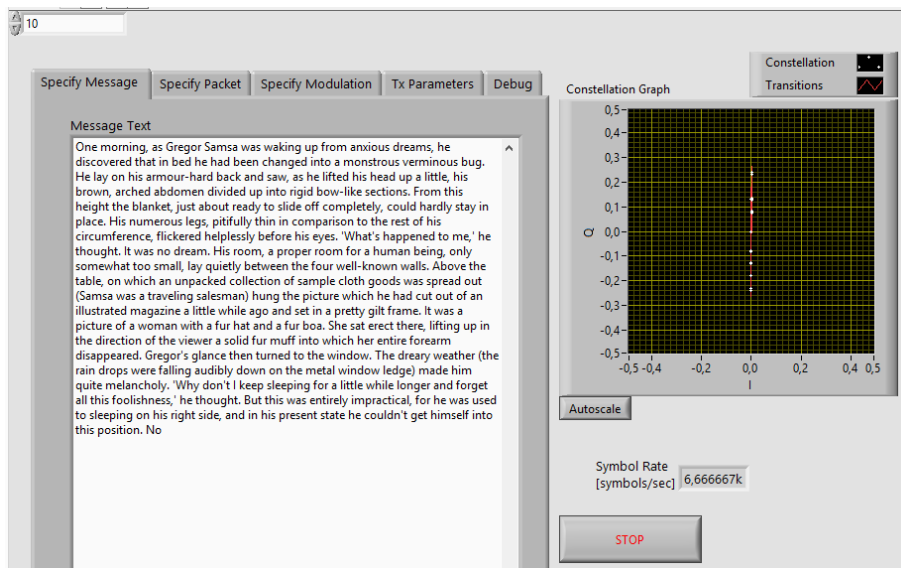
“One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his ey”.



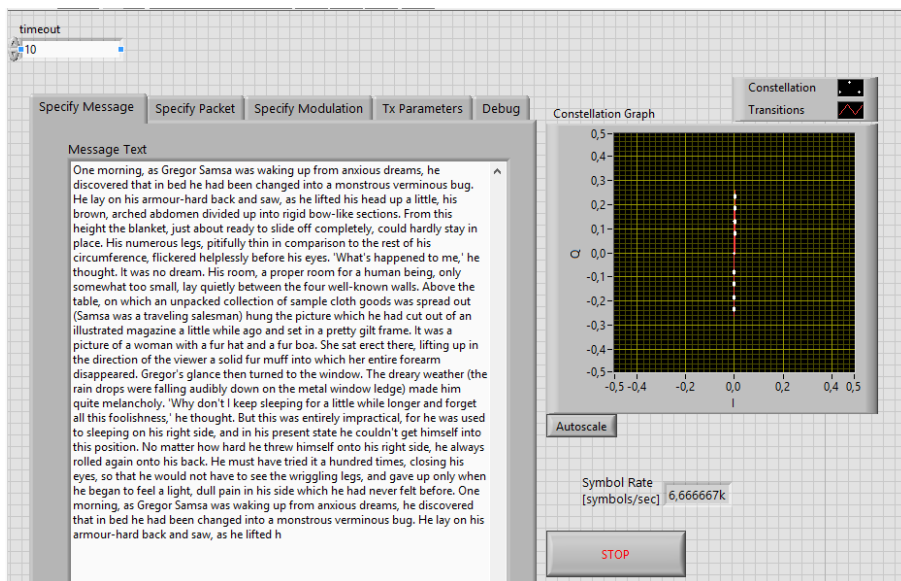
Gambar 4.11 Implementasi Pengiriman Teks 700 Karakter Dengan *Protocol*



Gambar 4.12 Implementasi Pengiriman Teks 1000 Karakter Dengan *Protocol*



Gambar 4.13 Implementasi Pengiriman Teks 1500 Karakter Dengan *Protocol*



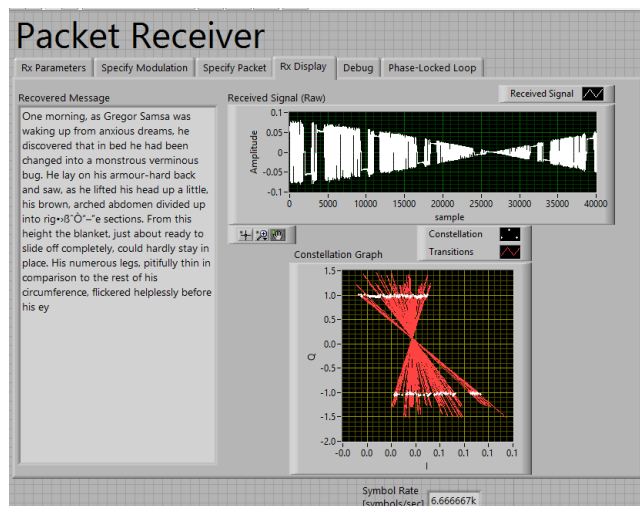
Gambar 4.14 Implementasi Pengiriman Teks 2000 Karakter Dengan *Protocol*

Gambar 4.10 tampilan hasil pengiriman teks 500 karakter, gambar 4.11 tampilan hasil pengiriman teks 700 karakter, gambar 4.12 tampilan hasil pengiriman teks 1000 karakter, gambar 4.13 tampilan hasil pengiriman teks 1500, dan gambar 4.14 tampilan hasil pengiriman teks 2000 karakter pada sisi pemancar dengan menggunakan *protocol* AX.25

pada *software labview* dengan parameter yang sudah di jelaskan pada sub bab 3.5.1.

#### 4.1.3 Hasil Implementasi Sistem Penerima Tanpa Protokol AX.25

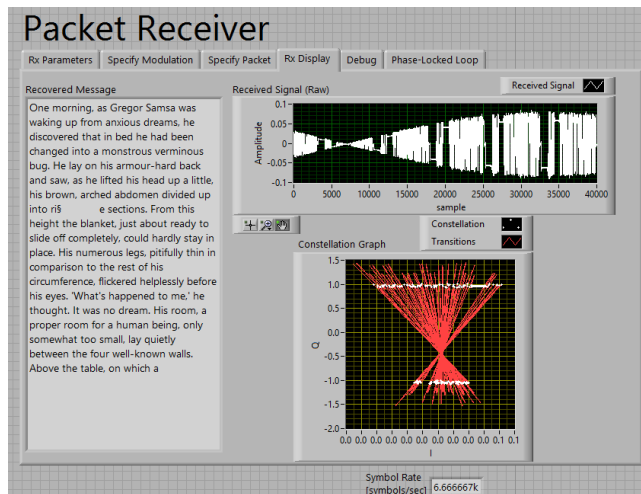
Pada penelitian ini di dapatkan hasil implementasi sistem komunikasi digital HF pada sisi penerima tanpa penggunaan *protocol AX.25* tidak semua teks berhasil diterima menggunakan *software labview* yang diintegrasikan ke USRP. Pengujian dilakukan dengan cara melakukan pengiriman data teks sebanyak 10 kali pada masing-masing variasi karakter. Variasi karakter yang digunakan yaitu 500, 700, 1000, 1500 dan 2000 karakter.



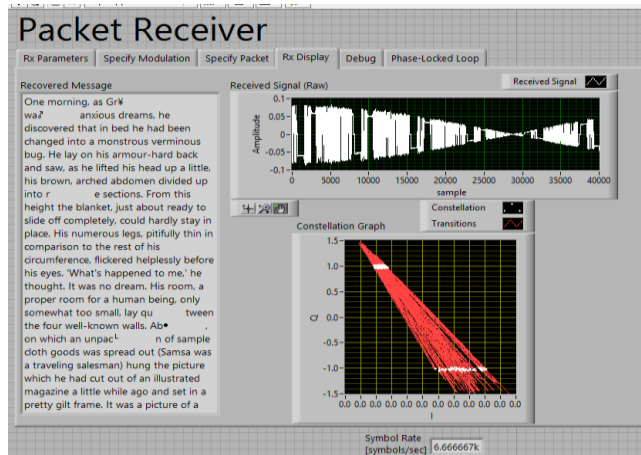
Gambar 4.15 Salah Satu Implementasi Penerimaan Teks 500 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa *Protocol*

Dengan rincian teks sebagai berikut :

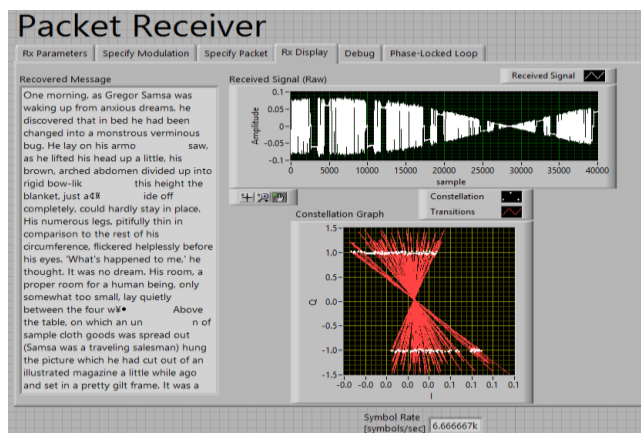
“One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rig•ß•^Ö“-“e sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his ey”.



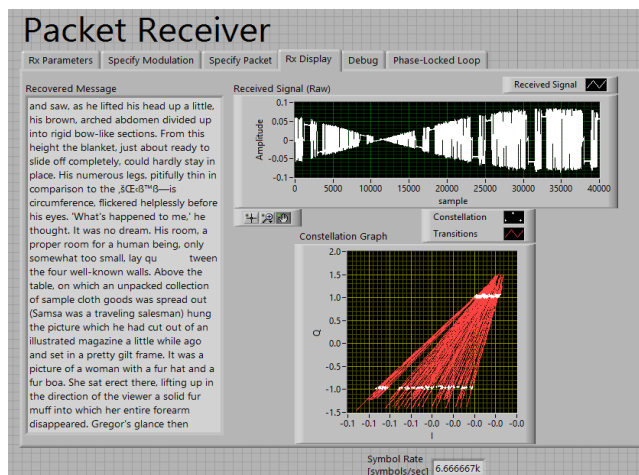
Gambar 4.16 Salah Satu Implementasi Penerimaan Teks 700 Karakter yang Tidak



Gambar 4.17 Salah Satu Implementasi Penerimaan Teks 1000 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa *Protocol*



Gambar 4.18 Salah Satu Implementasi Penerimaan Teks 1500 Karakter yang Tidak



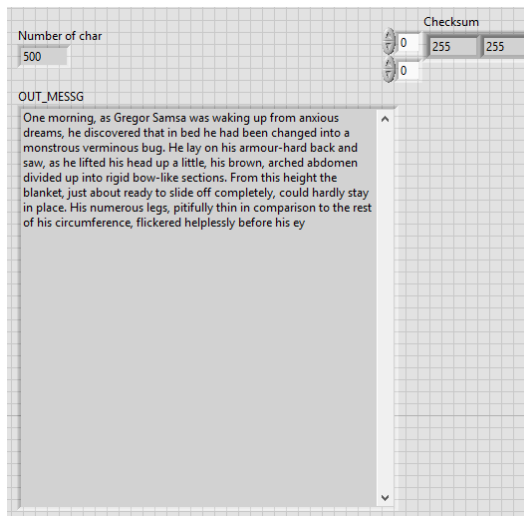
Gambar 4.19 Salah Satu Implementasi Penerimaan Teks 2000 Karakter yang Tidak Berhasil Diterima Lengkap Tanpa *Protocol*

Gambar 4.15 tampilan hasil penerimaan teks 500 karakter, gambar 4.16 tampilan hasil penerimaan teks 700 karakter, gambar 4.17 tampilan hasil penerimaan teks 1000 karakter, gambar 4.18 tampilan hasil penerimaan teks 1500, dan gambar 4.19 tampilan hasil penerimaan teks 2000 karakter pada sisi pemancar dengan menggunakan *protocol* AX.25 pada *software labview* dengan parameter yang sudah di jelaskan pada sub bab 3.5.2.

#### 4.1.4 Hasil Implementasi Sistem Penerima Dengan Protokol AX.25

Pada penelitian ini di dapatkan hasil implementasi sistem komunikasi digital HF pada sisi penerimaan dengan penggunaan *protocol* AX.25 teks berhasil diterima dengan lengkap menggunakan *software labview* yang diintegrasikan ke USRP. Pengujian dilakukan dengan cara melakukan pengiriman data teks sebanyak 10 kali pada masing-masing variasi karakter. Variasi karakter yang digunakan yaitu 500, 700, 1000, 1500 dan 2000 karakter.

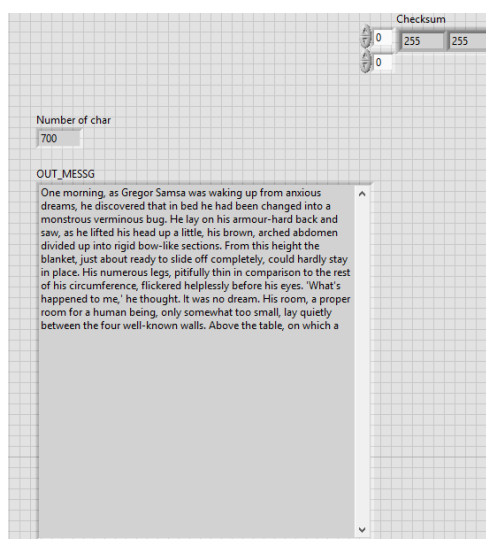




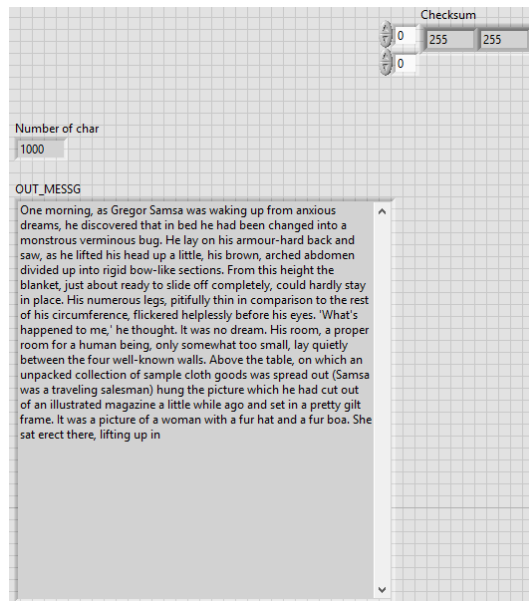
Gambar 4.20 Salah Satu Implementai Penerimaan Teks 500 Karakter Yang Berhasil Diterima Lengkap Dengan Protocol

Dengan rincian teks sebagai berikut :

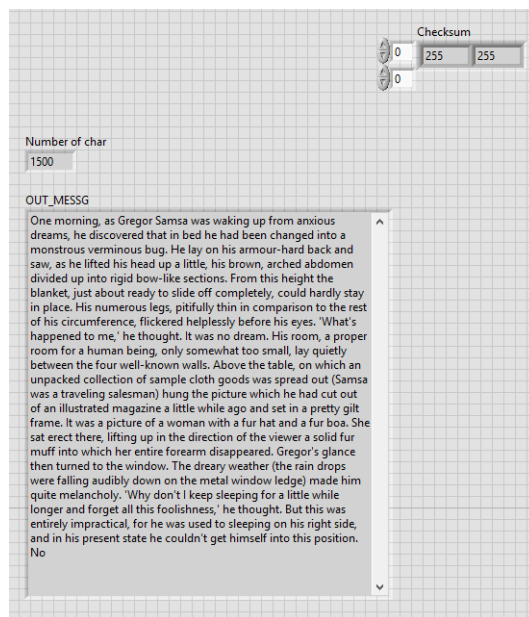
“One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his ey”.



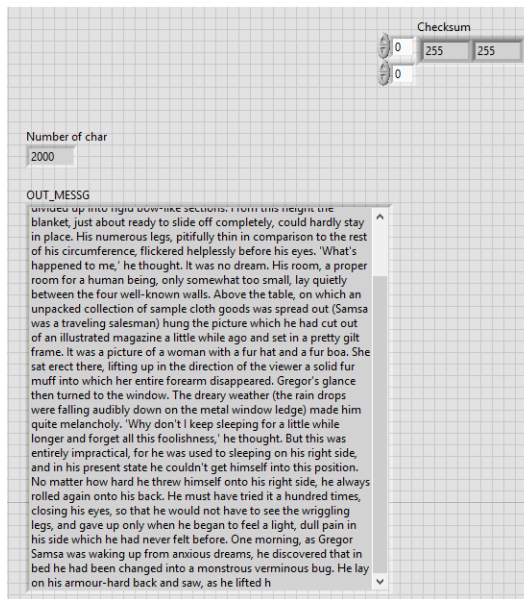
Gambar 4.21 Salah Satu Implementasi Penerimaan Teks 700 Karakter Yang Berhasil Diterima Lengkap Dengan *Protocol AX.25*



Gambar 4.22 Salah Satu Tampilan Hasil Penerimaan Teks 1000 Karakter Yang Berhasil Diterima Lengkap Dengan Penggunaan *Protocol AX.25*



Gambar 4.23 Salah Satu Implementasi Penerimaan Teks 1500 Karakter Yang Berhasil Diterima Lengkap Dengan *Protocol*



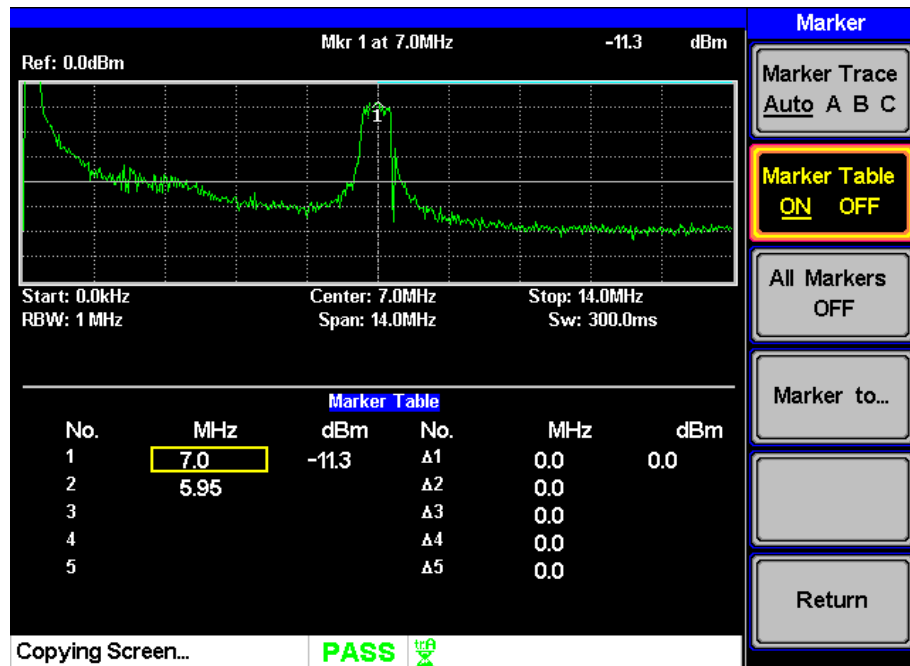
Gambar 4.24 Salah Satu Implementasi Penerimaan Teks 2000 Karakter Yang Berhasil Diterima Lengkap Dengan *Protocol*

Gambar 4.20 tampilan hasil penerimaan teks 500 karakter, gambar 4.21 tampilan hasil penerimaan teks 700 karakter, gambar 4.22 tampilan hasil penerimaan teks 1000 karakter, gambar 4.23 tampilan hasil penerimaan teks 1500, dan gambar 4.24 tampilan hasil penerimaan teks 2000 karakter pada sisi penerima dengan menggunakan *protocol AX.25* pada *software labview* berhasil diterima lengkap dengan parameter yang sudah di jelaskan pada sub bab 3.5.2.

## 4.2 Analisis Sistem Pemancar

### 4.2.1 Pembangkitan Sinyal Pada Pemancar Terhadap Frekuensi

Pada penelitian sistem komunikasi digital HF kali ini frekuensi yang digunakan bekerja pada frekuensi 7 MHz sesuai dengan desain frekuensi kerja, dan di dapatkan nilai pengukuran daya sinyal -11.3 dBm. Dapat digambarkan pada gambar 4.25 sebagai berikut.



Gambar 4.25 Hasil Pembangkitan Sinyal Frekuensi Kerja

#### 4.2.2 Output Power Amplifier

Dengan menggunakan tegangan 24 Volt pada DC supply, Output power amplifier yang dihasilkan pada penelitian ini adalah sekitar 3 Watt pada pemancar (TX), dapat digambarkan pada gambar 4.26 sebagai berikut.



Gambar 4.26 Daya Transmitter

### 4.2.3 Pengiriman Data Teks Berdasarkan Penggunaan Tanpa *Protocol* dan Dengan *Protocol AX.25*

Dalam penelitian sistem komunikasi digital HF ini dilakukan pengujian pengiriman data teks. Pengiriman data teks berdasarkan penggunaan tanpa *protocol* dan dengan *protocol AX.25*. Pengujian pengiriman data teks pada masing-masing variasi jumlah karakter sebanyak 10 kali pengiriman data teks. Jumlah karakter dalam variasi pengiriman data teks dapat ditunjukkan pada tabel 4.1 sebagai berikut.

Tabel 4.1 Parameter Variasi Jumlah Karakter Pengiriman

No	Jumlah Karakter	Pengiriman Data Teks (Kali)	
		Tanpa <i>Protocol AX.25</i>	Dengan <i>Protocol AX.25</i>
1	500 Karakter	10	10
2	700 Karakter	10	10
3	1000 Karakter	10	10
4	1500 Karakter	10	10
5	2000 Karakter	10	10

## 4.3 Analisis Sistem Penerima

### 4.3.1 Analisis Hasil *Signal to Noise Ratio* (SNR) Rata-Rata Berdasarkan Penggunaan Tanpa *Protocol* dan Dengan *Protocol AX.25*

Berdasarkan hasil pengujian penelitian yang dilakukan sebanyak 10 kali pengujian penerimaan maka didapatkan nilai SNR rata-rata berdasarkan penggunaan tanpa *protocol* dan dengan *protocol AX.25* dengan variasi jumlah karakter dapat ditunjukkan pada tabel 4.2.

Tabel 4.2 Hasil Nilai SNR Rata-Rata Penggunaan Tanpa *Protocol* dan Dengan *Protocol AX.25*

Jumlah Karakter	SNR Rata-Rata (dB)	
	Tanpa <i>Protocol AX.25</i>	Dengan <i>Protocol AX.25</i>
500 Karakter	20.22	26.63
700 Karakter	23.77	27.67
1000 Karakter	21.65	26.19
1500 Karakter	21.55	26.21
2000 Karakter	21.72	27.88

Berdasarkan hasil penelitian, dilakukan perbandingan sistem tanpa dan dengan *protocol AX.25* dengan jumlah karakter yang bervariasi. Maka penggunaan *protocol AX.25* dapat memberikan *zero error* pada sistem komunikasi digital HF, dan tidak memberikan peningkatan kualitas nilai SNR rata-rata yang signifikan terhadap penggunaan dengan *protocol AX.25*.

#### 4.3.2 Analisis Hasil Bit Error Rate (BER) Rata-Rata Tanpa Protocol dan Dengan Protocol AX.25

Berdasarkan hasil pengujian penelitian yang dilakukan sebanyak 10 kali pengujian penerimaan maka didapatkan nilai BER rata-rata berdasarkan penggunaan tanpa *protocol* dan dengan *protocol AX.25* dengan variasi jumlah karakter dapat ditunjukkan pada tabel 4.3.

Tabel 4.3. Hasil BER Rata-Rata Penggunaan Tanpa Protokol AX.25 dan Dengan Protokol AX.25

Jumlah Karakter	BER Rata-Rata	
	Tanpa <i>Protocol AX.25</i>	Dengan <i>Protocol AX.25</i>
500 Karakter	0.09544	0
700 Karakter	0.00489	0
1000 Karakter	0.14716	0
1500 Karakter	0.19848	0
2000 Karakter	0.22698	0

Berdasarkan hasil penelitian, dilakukan perbandingan sistem tanpa dan dengan *protocol AX.25* dengan jumlah karakter yang bervariasi. Maka di dapatkan trend peningkatan nilai BER rata-rata pada sistem tanpa *protocol AX.25*, akan tetapi penggunaan sistem dengan *protocol AX.25* dapat memberikan *zero error* pada sistem komunikasi digital HF.

#### 4.3.3 Analisis Hasil Character Error Rate (CER) Rata-Rata Tanpa Protocol dan Dengan Protocol AX.25

Berdasarkan hasil pengujian penelitian yang dilakukan sebanyak 10 kali pengujian penerimaan maka didapatkan nilai CER rata-rata

berdasarkan penggunaan tanpa *protocol* dan dengan *protocol AX.25* dengan variasi jumlah karakter dapat ditunjukkan pada tabel 4.4.

Tabel 4.4. Hasil CER Rata-Rata Penggunaan Tanpa *Protocol AX.25* dan Dengan *Protocol AX.25*

Jumlah Karakter	CER Rata-Rata	
	Tanpa <i>Protocol AX.25</i>	Dengan <i>Protocol AX.25</i>
500 Karakter	0.0134	0
700 Karakter	0.00743	0
1000 Karakter	0.0301	0
1500 Karakter	0.1072	0
2000 Karakter	0.0133	0

Berdasarkan hasil penelitian, dilakukan perbandingan sistem tanpa dan dengan *protocol AX.25* dengan jumlah karakter yang bervariasi. Maka di dapatkan trend peningkatan nilai CER rata-rata pada sistem tanpa *protocol AX.25*, akan tetapi penggunaan sistem dengan *protocol AX.25* dapat memberikan *zero error*. Penurunan nilai CER rata-rata tidak berpengaruh terhadap peningkatan nilai BER rata-rata pada sistem tersebut.

#### 4.3.4 Analisis Hasil *Frame Error Rate (FER)* Rata-Rata Tanpa Protokol dan Dengan Protokol AX.25

Berdasarkan hasil pengujian penelitian yang dilakukan sebanyak 10 kali pengujian penerimaan maka didapatkan nilai FER rata-rata berdasarkan penggunaan tanpa *protocol* dan dengan *protocol AX.25* dengan variasi jumlah karakter dapat ditunjukkan pada tabel 4.5.

Tabel 4.5 Hasil FER Rata-Rata Penggunaan Tanpa *Protocol AX.25* dan Dengan *Protocol AX.25*

Jumlah Karakter	FER Rata-Rata	
	Tanpa <i>Protocol AX.25</i>	Dengan <i>Protocol AX.25</i>
500 Karakter	0.25	0
700 Karakter	0.1	0
1000 Karakter	0.25	0
1500 Karakter	0.3667	0
2000 Karakter	0.2125	0

Berdasarkan hasil penelitian, dilakukan perbandingan sistem tanpa dan dengan *protocol AX.25* dengan jumlah karakter yang bervariasi. Maka di dapatkan trend peningkatan nilai FER rata-rata pada sistem tanpa *protocol AX.25*, akan tetapi penggunaan sistem dengan *protocol AX.25* dapat memberikan *zero error*. Penurunan nilai FER rata-rata tidak berpengaruh terhadap peningkatan nilai BER rata-rata pada sistem tersebut.

#### 4.4 Sintesis

Dari hasil penelitian sistem komunikasi digital HF berdasarkan penggunaan tanpa *protocol* dan dengan *protocol AX.25* di ketahui hasil implementasi sistem tanpa *protocol* dan dengan *protocol AX.25* pada sisi pemancar berhasil melakukan pengiriman data teks dengan variasi jumlah karakter yang berbeda.

Hasil implementasi sistem penerima tanpa penggunaan *protocol AX.25* belum bisa di terima lengkap data teks yang dikirimkan dari pemancar dengan variasi jumlah karakter yang berbeda. Sedangkan, hasil implementasi sistem penerima dengan penggunaan *protocol AX.25* berhasil di terima dengan lengkap data teks yang dikirimkan dari pemancar dengan variasi jumlah karakter yang berbeda.

Penelitian ini di desain menggunakan frekuensi kerja 7 MHz pada sisi pemancar dan penerima berdasarkan tanpa penggunaan *protocol* dan dengan penggunaan *protocol AX.25* sesuai dengan peraturan IAIR. Konfigurasi perangkat yang digunakan pada sistem ini tidak ada perbedaan pada sisi pemancar dan penerima berdasarkan tanpa penggunaan *protocol* dan dengan penggunaan *protocol AX.25*. Selain itu, dengan *input DC power supply* sebesar 24 volt hanya menghasilkan *output* daya pancar sebesar 3 watt pada sisi pemancar berdasarkan tanpa penggunaan *protocol* dan dengan penggunaan *protocol AX.25*.

Pengujian pengiriman data teks dilakukan sebanyak 10 kali pada masing-masing variasi jumlah karakter yang berbeda-beda. Variasi jumlah karakter yang digunakan yaitu 500, 700, 1000, 1500 dan 2000 karakter.



Berdasarkan hasil pengujian, nilai SNR rata-rata berdasarkan penggunaan tanpa *protocol* dan penggunaan dengan *protocol* AX.25 dapat memberikan *zero error* akan tetapi tidak ada perubahan nilai SNR rata-rata yang signifikan. Berdasarkan hasil pengujian, nilai BER rata-rata berdasarkan penggunaan tanpa *protocol* AX.25 terjadi trend peningkatan nilai BER rata-rata, sedangkan penggunaan dengan *protocol* AX.25 dapat memberikan *zero error* pada sistem. Berdasarkan hasil pengujian, nilai CER rata-rata berdasarkan penggunaan tanpa *protocol* AX.25 terjadi trend peningkatan nilai CER rata-rata, sedangkan penggunaan dengan *protocol* AX.25 dapat memberikan *zero error* pada sistem. Akan tetapi, penurunan nilai CER rata-rata tidak berpengaruh terhadap peningkatan nilai BER rata-rata. Berdasarkan hasil pengujian, nilai FER rata-rata berdasarkan penggunaan tanpa *protocol* AX.25 terjadi trend peningkatan nilai FER rata-rata, sedangkan penggunaan dengan *protocol* AX.25 dapat memberikan *zero error* pada sistem. Akan tetapi, penurunan nilai FER rata-rata tidak berpengaruh terhadap peningkatan nilai BER.

## BAB 5

### KESIMPULAN

#### 5.1 KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut:

1. Desain sistem komunikasi digital HF berdasarkan penggunaan tanpa *protocol AX.25* pada sisi pemancar berhasil di desain pada *platform SDR* dengan mendesain diantaranya *Input* data teks maksimal 2000 karakter, *Tx IQ Rate*, *Tx Frequency*, *Tx Gain*, *Tx Antenna*, Diagram Konstelasi, *Guard Bits*, *Sync Bit*, *Message Bits*, Modulasi, *Samples per Symbol*, *Tx Filter*, dan *Roll Of Factor/Alpha*.
2. Desain sistem komunikasi digital HF berdasarkan penggunaan dengan *protocol AX.25* pada sisi pemancar berhasil di desain pada *platform SDR* dengan mendesain diantaranya *Input* data teks maksimal 2000 karakter, *Tx IQ Rate*, *Tx Frequency*, *Tx Gain*, *Tx Antenna*, Diagram Konstelasi, *Guard Bits*, *Sync Bit*, *Message Bits*, Modulasi, *Samples per Symbol*, *Tx Filter*, *Roll Of Factor/Alpha*, *Encoder*, *STR\_From\_Callsign*, *STR\_To\_Callsign*, *FCS Tx*, *Message Bitstream*, *Control*, dan *PID*.
3. Desain sistem komunikasi digital HF berdasarkan penggunaan tanpa *protocol AX.25* pada sisi penerima berhasil di desain pada *platform SDR* dengan mendesain diantaranya *Output* data teks maksimal 2000 karakter, *Rx IQ Rate*, *Rx Frequency*, *Rx Gain*, *Rx Antenna*, Diagram Konstelasi, *Guard Bits*, *Sync Bit*, *Message Bits*, PLL, Modulasi, *Samples per Symbol*, *Rx Filter*, *Signal Power*, *Noise Power* dan *Roll Of Factor/Alpha*.
4. Desain sistem komunikasi digital HF berdasarkan penggunaan dengan *protocol AX.25* pada sisi penerima berhasil di desain pada *platform SDR* dengan mendesain diantaranya *Output* data teks maksimal 2000 karakter, *Rx IQ Rate*, *Rx Frequency*, *Rx Gain*, *Rx*

*Antenna, Diagram Konstelasi, Guard Bits, Sync Bit, Message Bits, Modulasi, Samples per Symbol, Rx Filter, Signal Power, Noise Power, PLL, Roll Of Factor/Alpha, Decoder, FCS Rx, Check Sum, Acquisition Duration, Number of Character dan Number of Packet to Expect.*

5. Hasil implementasi sistem komunikasi digital HF pada sisi pemancar tanpa penggunaan *protocol AX.25* berhasil dikirimkan dengan baik dengan *platform SDR* dengan variasi pengiriman jumlah karakter yang berbeda-beda.
6. Hasil implementasi sistem komunikasi digital HF pada sisi pemancar dengan penggunaan *protocol AX.25* berhasil dikirimkan dengan baik dengan *platform SDR* dengan variasi pengiriman jumlah karakter yang berbeda-beda.
7. Hasil implementasi sistem penerima tanpa penggunaan *protocol AX.25* belum bisa di terima lengkap data teks yang dikirimkan dari pemancar dengan variasi jumlah karakter yang berbeda.
8. hasil implementasi sistem penerima dengan penggunaan *protocol AX.25* berhasil di terima dengan lengkap data teks yang dikirimkan dari pemancar dengan variasi jumlah karakter yang berbeda.
9. Penelitian ini di desain menggunakan frekuensi kerja 7 MHz pada sisi pemancar dan penerima berdasarkan tanpa penggunaan *protocol* dan dengan penggunaan *protocol AX.25* sesuai dengan peraturan IAIR.
10. Konfigurasi perangkat yang digunakan pada sistem ini tidak ada perbedaan pada sisi pemancar dan penerima berdasarkan tanpa penggunaan *protocol* dan dengan penggunaan *protocol AX.25*.
11. Dengan *input DC power supply* sebesar 24 volt hanya menghasilkan *output* daya pancar sebesar 3 watt pada sisi pemancar berdasarkan tanpa penggunaan *protocol* dan dengan penggunaan *protocol AX.25*.
12. Pengujian pengiriman data teks dilakukan sebanyak 10 kali pada masing-masing variasi jumlah karakter yang berbeda-beda. Variasi

jumlah karakter yang digunakan yaitu 500, 700, 1000, 1500 dan 2000 karakter.

13. Berdasarkan hasil pengujian, nilai SNR rata-rata berdasarkan penggunaan tanpa *protocol* dan penggunaan dengan *protocol AX.25* dapat memberikan *zero error* akan tetapi tidak ada perubahan nilai SNR rata-rata yang signifikan.
14. Berdasarkan hasil pengujian, nilai BER rata-rata berdasarkan penggunaan tanpa *protocol AX.25* terjadi trend peningkatan nilai BER rata-rata, sedangkan penggunaan dengan *protocol AX.25* dapat memberikan *zero error* pada sistem.
15. Berdasarkan hasil pengujian, nilai CER rata-rata berdasarkan penggunaan tanpa *protocol AX.25* terjadi trend peningkatan nilai CER rata-rata, sedangkan penggunaan dengan *protocol AX.25* dapat memberikan *zero error* pada sistem. Akan tetapi, penurunan nilai CER rata-rata tidak berpengaruh terhadap peningkatan nilai BER rata-rata.
16. Berdasarkan hasil pengujian, nilai FER rata-rata berdasarkan penggunaan tanpa *protocol AX.25* terjadi trend peningkatan nilai FER rata-rata, sedangkan penggunaan dengan *protocol AX.25* dapat memberikan *zero error* pada sistem. Akan tetapi, penurunan nilai FER rata-rata tidak berpengaruh terhadap peningkatan nilai BER.

## 5.2 SARAN

Dalam pengembangan sistem komunikasi digital HF selanjutnya dapat disarankan untuk penelitian selanjutnya diantaranya:

1. Jika memungkinkan dapat dilakukan pengujian sistem komunikasi digital HF jarak jauh.
2. Jika memungkinkan dapat dilakukan pembuatan desain sistem komunikasi digital HF MIMO.
3. Jika memungkinkan implementasi desain sistem komunikasi digital HF MIMO.

4. Jika memungkinkan dilakukan uji coba sistem komunikasi digital HF MIMO.

## DAFTAR PUSTAKA

- [1] L. F. McNamara, *The Ionosphere: Communications, Surveillance, and Direction Finding*, Krieger, USA, 1991.
- [2] Y. Erhel, D. Lemur, M. Oger, J. Le Mason, F. Marie, "Evaluation Of Ionospheric HF MIMO Channels: Two Complementary Circular Polarizations Reduce Correlation", di *IEEE Ant. and Prop. Mag.*, Vol. 58, No. 6, 2016, pp. 38-48.
- [3] Walden M.C., "High-Frequency Near Vertical Incidence Skywave Propagation: Findings Associated With The 5 MHz Experiment", di *IEEE Ant. and Prop. Mag.*, Vol. 58, No. 6, 2016, pp. 16-28.
- [4] S. L. Manalu, G. Hendratoro, A. Mauludiyanto, "Design of Measurement System for HF MIMO NVIS Channels," *Int. Conf. on Information Tech., Computer, and Electrical Engineering*, Semarang, Indonesia, 2017.
- [5] B. A. Witvliet and R. M. Alsina-pages., "Radio Communication via Near Vertical Incidence Skywave propagation: An overview," in *Springer telecomm. System*, 2017.
- [6] Mukti P. H., I. Kurniawati, F. Oktaviansyah, A.D. Adhitya, N. Rachmadani, R. Corputty, G. Hendratoro, T. Fukusako, "Software-Defined Radio Based Channel Measurement System of Wideband HF Communication System in Low-Latitude Region", *World Academy of Science, Engineering and Technology International Journal of Electronics and Communication Engineering*, Vol. 8, No. 5, 2014, pp. 850-853.
- [7] National Instrument, *LabVIEW User Manual*, National Instruments Corporation, USA, 2003.
- [8] Davidovici, S., Ph.D., R. Simpson, Ph.D., "Detectability of Packet Communications Using Variants of the X.25 Protocol", *IEEE Conference on Military Communications*, Monterey, CA, USA, 1990.
- [9] R.R. Parry, "AX.25 [data link layer protocol for packet radio networks]", in *IEEE Potentials*, vol.16, No.3, 1997, pp. 14-16.

- [10] Ndao P.M., Y. Erhel, D. Lemur, J. Le Masson," Design of a high-frequency (3-30 MHz) multiple-input-multiple-output system resorting to polarisation diversity, in IET Microwaves, Ant. and Prop. Vol. 5, No. 11, 2011, pp. 1310-1318.
- [11] Australian Government,"*IPS and Space Services, Introduction to HF Radio Propagation*", di Sidney, Australia, 2007.
- [12] Suhartini, S,"*Komunikasi Radio High Frequency Jarak Dekat*," di LAPAN, Indonesia. 2011.
- [13] K. Davies, *Ionospheric Radio*. London, UK: Peter Peregrinus, 1990.
- [14] B.A Witvliet, E. Van Maanen, G.J. Petersen, A.J. Westenberg M.J. Bentum, "The Importance of Circular for Diversity Reception and MIMO in NVIS propagation," *Netherlands Antenna research framework*, 2014.
- [15] Sklar, B. *Digital Communications Fundamental and Applications, Second Edition*. New Jersey: Prentice Hall. 2001.
- [16] J. G. Proakis and M. Salehi, *Digital Communications*, McGraw-Hill, 2007.
- [17] Hassani, S.A,"*Design and Implementation of FPGA-Based Multi-Rate BPSK-QPSK Modern with Focus on Carrier Recovery and Time Synchronization*", di Tampere University of Technology, Finland. 2016.
- [18] Beech, Willian A., Douglas E. Nielsen, and Jack Taylor, AX.25 Link Access Protocol for Amateur Packet Radio Version 2.2, Greg Jones, 1998.
- [19] HSU, Hwei P, *Analog and Digital Communication*, Schaum's Outline Second Edition, McGraw-Hill, 2005.
- [20] Best, R.E, *Phase-Locked Loops Design, Simulation, and Applications Fifth Edition*. McGraw-Hill. 2003.
- [21] Lyb, K.J.W.,"*A Technique for Calculating Ionospheric Doppler Shifts from Standard Ionograms Suitable for Scientific, HF Communication, and OTH Radar Applications*,"di Radio Science,Vol. 444, RS6002, 2009.
- [22] <http://194.75.38.69/pages/pdfs/an95003.pdf>. Diakses pada tanggal 19 Mei 2019.
- [23] Stallings, William. *Data and Computer Communications*. Pearson. 2007.
- [24] Balanis C.A., *Antenna Theory Analysis and Design*. USA: John Wiley, 2005
- [25] Spectrum Requirements for the Amateur and Amateur-satellite Services. International Amateur Radio Union. Revised November 2014.

- [26] Organisasi Radio Amatir Republik Indonesia, 2009, Keputusan Ketua Umum Organisasi Amatir Radio Indonesia Nomor: KEP-065/OP/KU/2009, "Judul Keputusan".
- [27] Menteri Komunikasi dan Informatika Republik Indonesia, 2009, Peraturan Menteri No.33 Tahun 2009 tentang Penyelenggaraan Amatir Radio.
- [28] Recommendation ITU-R F.339-8 (02/2013). Bandwidth, signal-to-noise ratios and fading allowances in HF fixed and land mobile radiocommunication systems.



*Halaman ini sengaja dikosongkan*

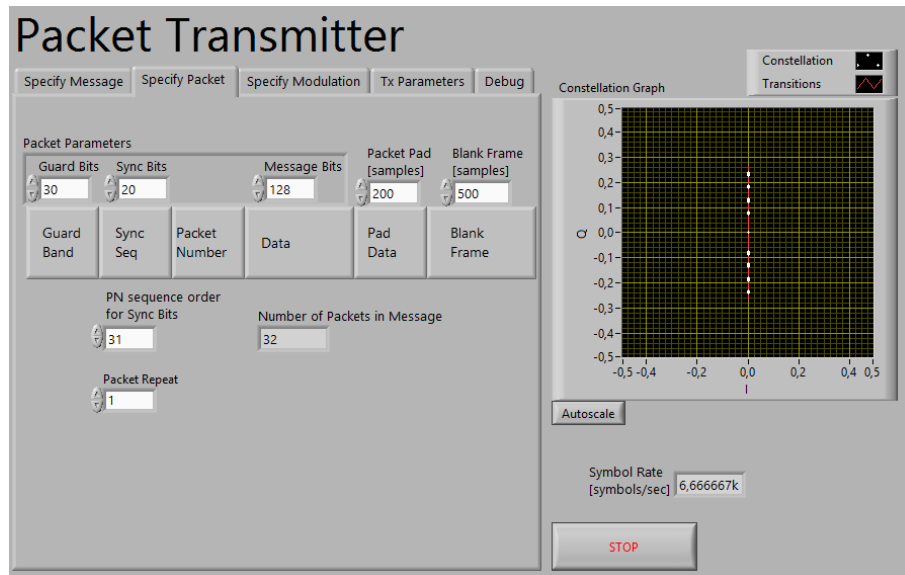
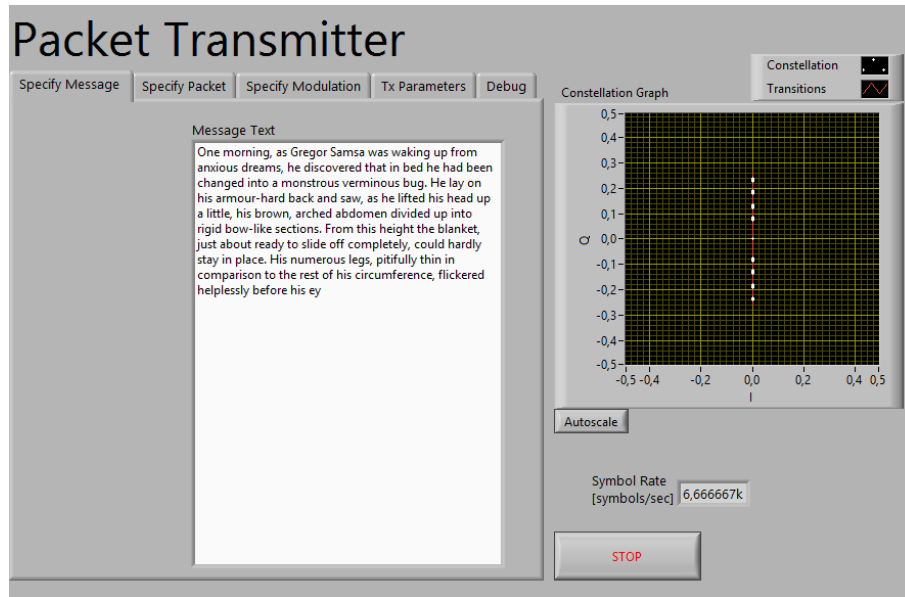
# LAMPIRAN 1

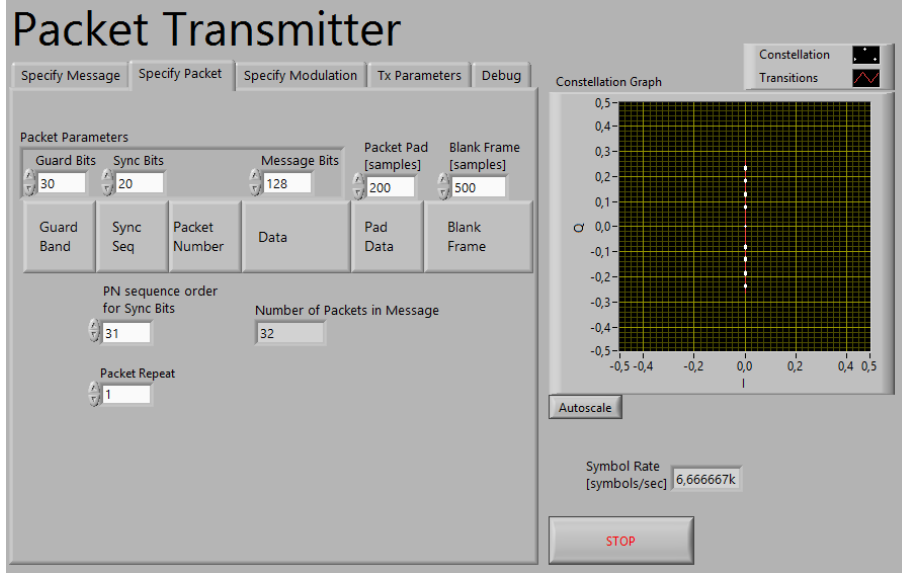
## Pengujian Labview 500, 700, 1000, 1500, 2000 di Software Labview

### 1. Contoh Pengujian Labview Tx dan Rx 500 Karakter Tanpa Protocol AX.25 dan Dengan Protocol AX.25

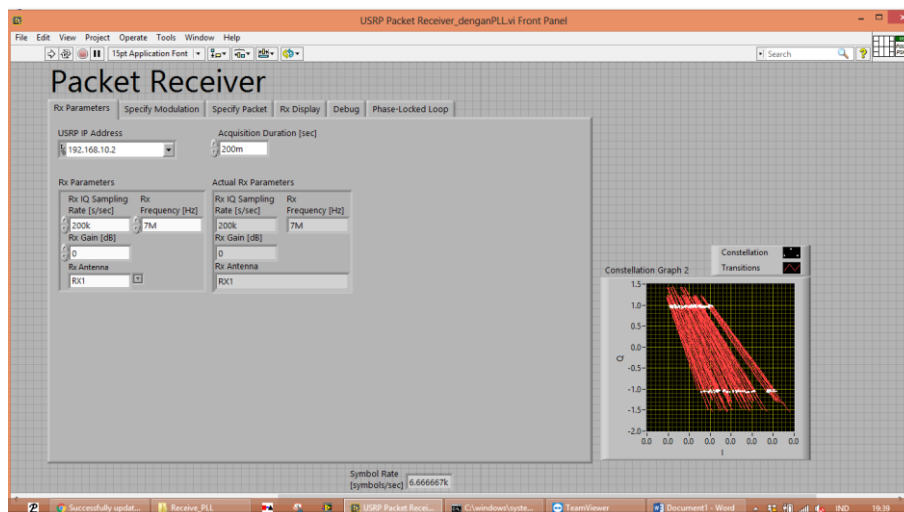
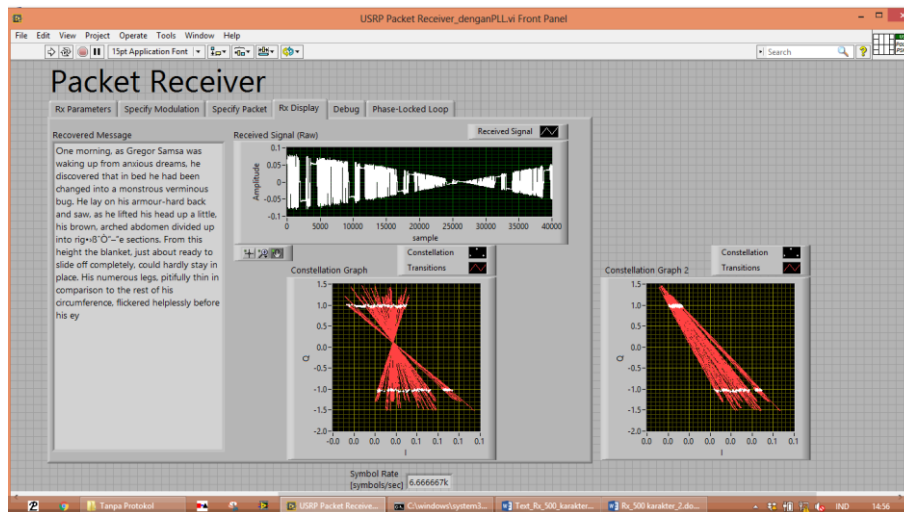
#### Tanpa Protocol AX.25 :

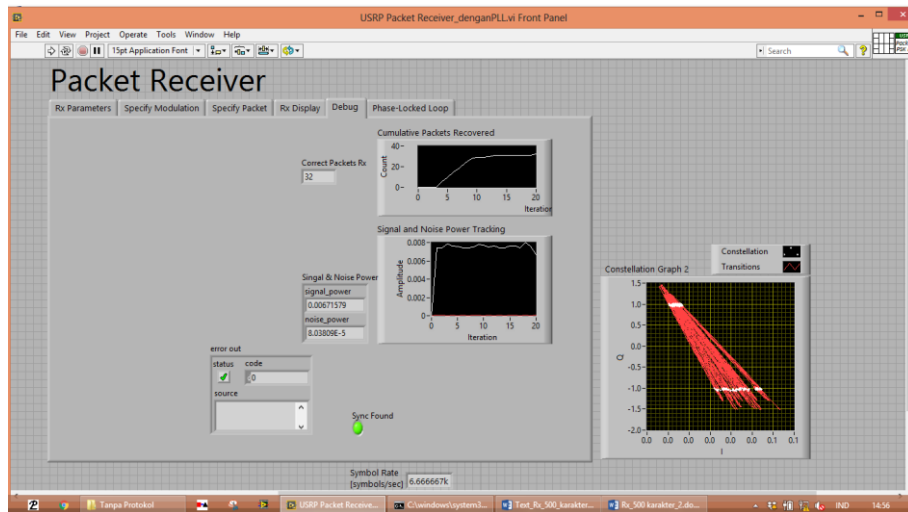
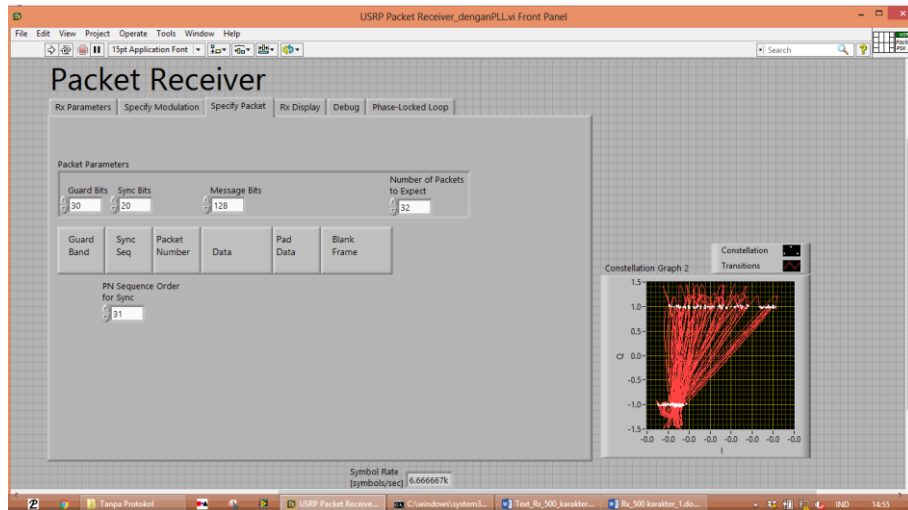
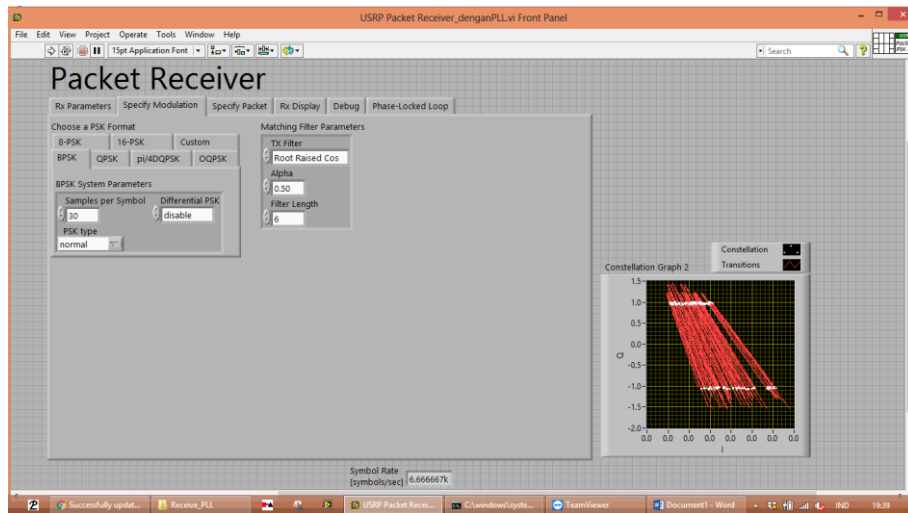
#### Transmitter :



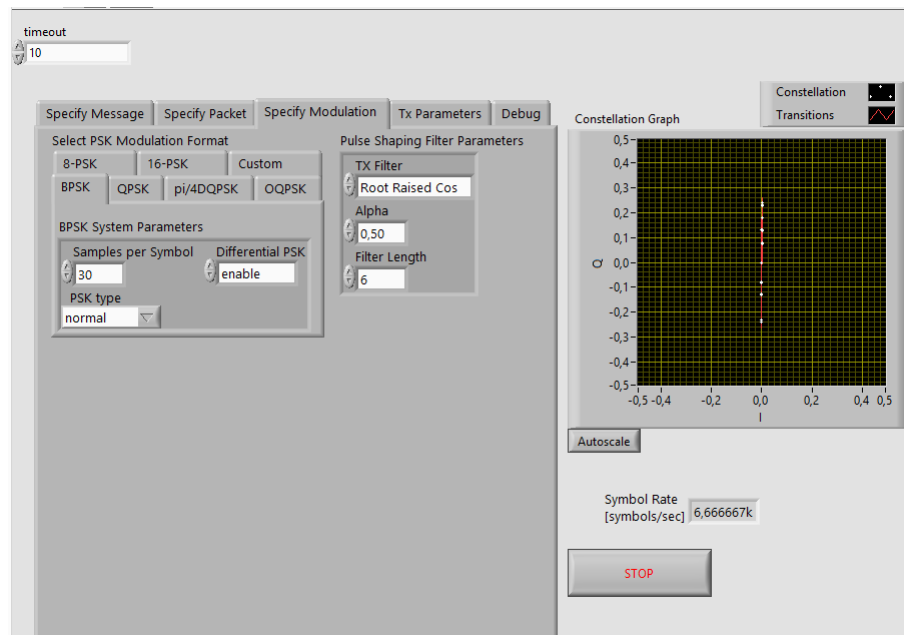
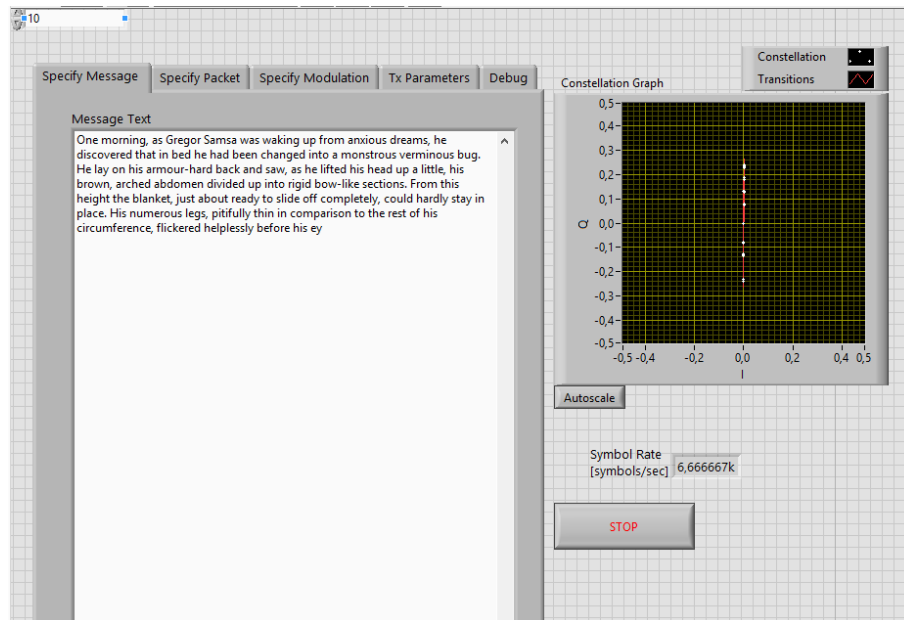


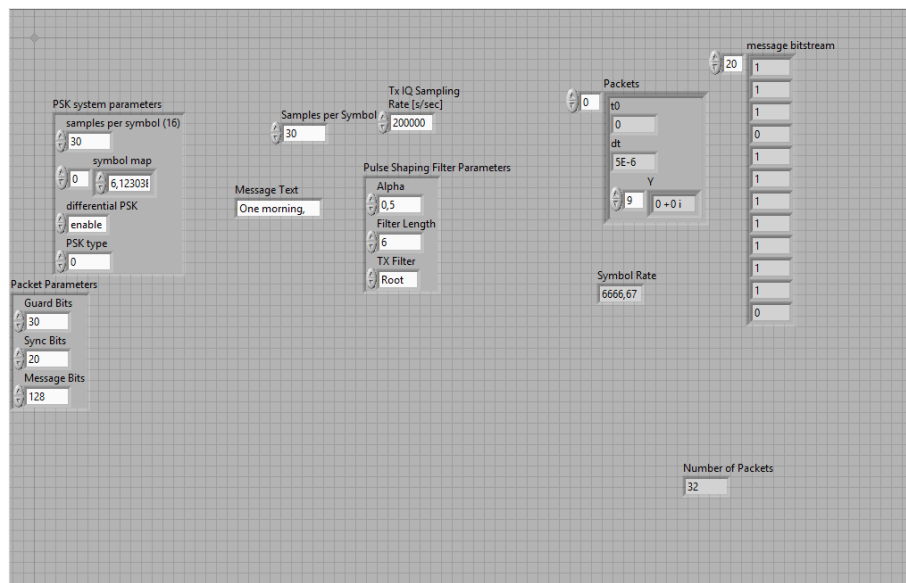
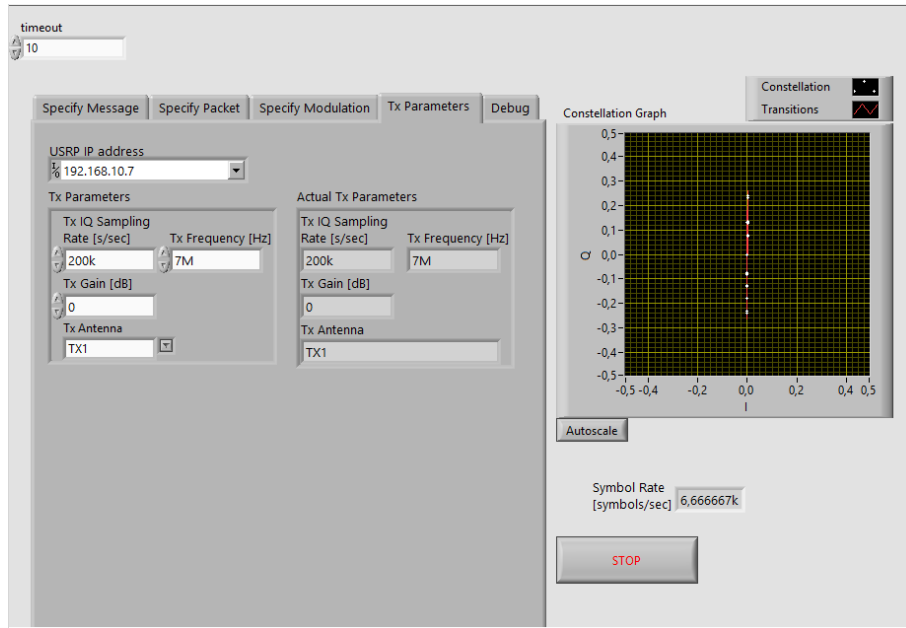
**Receiver :**





## Dengan Protocol AX.25: Transmitter :





passed ●

STR_FROM_CALLSIGN	STR_TO_CALLSIGN	CONTROL	FCS
YG3EGY	YG3GZ	03	F0 13115

text  
One morning, as Gregor Samsa

message bitstream

23	0
	1
	1
	1
	1
	1
	1
	1
	0
	1
	1
	1

File Length (bits)  
16000

Packet Message Length (bits)  
128

Number of Packets  
32

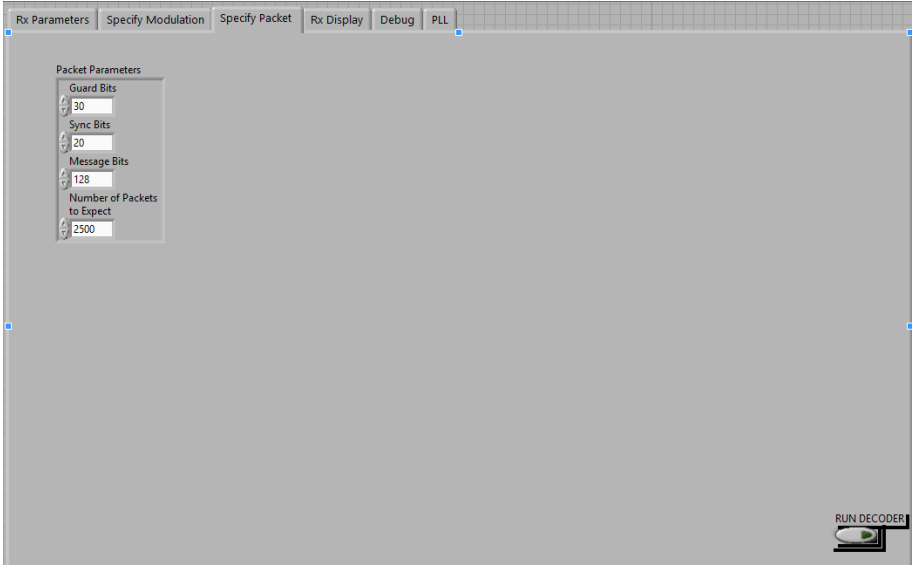
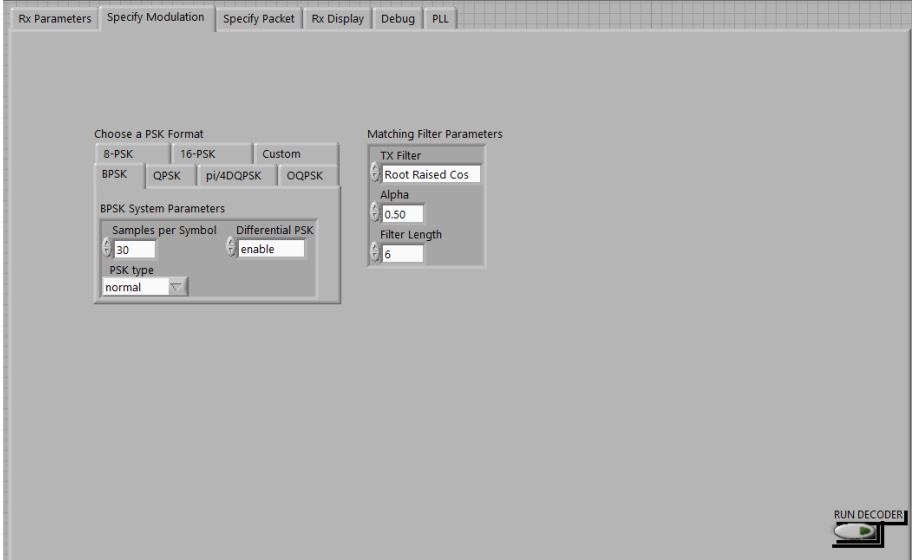
**Receiver :**

Rx Parameters   Specify Modulation   Specify Packet   Rx Display   Debug   PLL

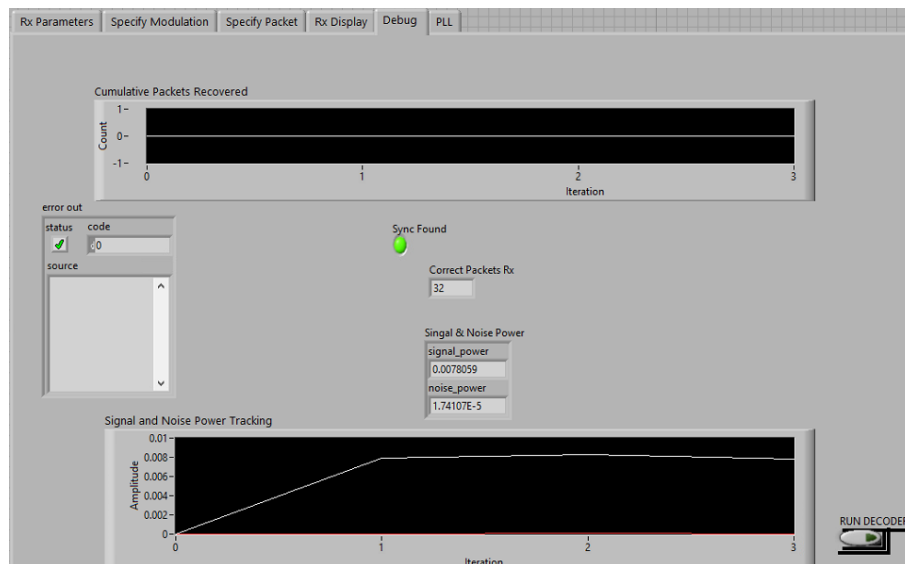
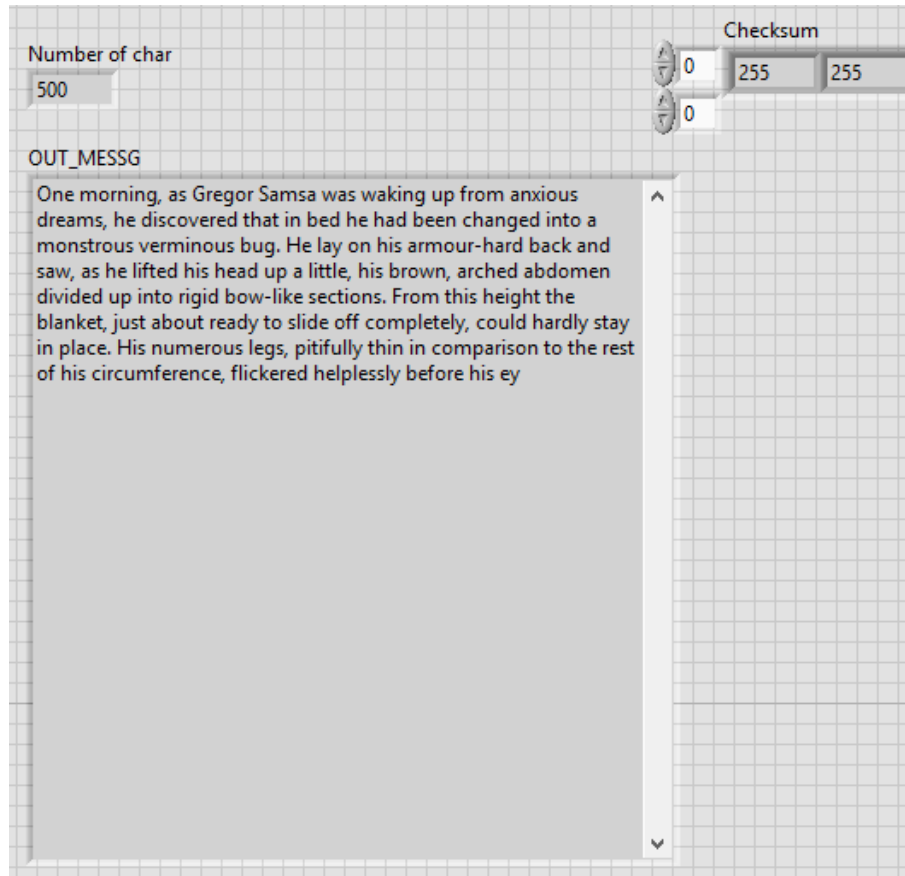
USRP IP Address: 192.168.10.2   Acquisition Duration [sec]: 200m

Rx Parameters		Actual Rx Parameters	
Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]	Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]
200k	7M	200k	7M
Rx Gain [dB]: 0		Rx Gain [dB]: 0	
Rx Antenna: RX1		Rx Antenna: RX1	

RUN DECODER







## 2. Contoh Pengujian Labview Tx dan Rx 700 Karakter Tanpa Protocol AX.25 dan Dengan Protocol AX.25

### Tanpa Protocol AX.25 :

Transmitter :

# Packet Transmitter

Specify Message | Specify Packet | Specify Modulation | Tx Parameters | Debug

Message Text

One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which a

Constellation Graph

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

# Packet Transmitter

Specify Message | Specify Packet | Specify Modulation | Tx Parameters | Debug

Packet Parameters

Guard Bits	Sync Bits	Message Bits	Packet Pad [samples]	Blank Frame [samples]
30	20	128	200	500

Guard Band	Sync Seq	Packet Number	Data	Pad Data	Blank Frame
------------	----------	---------------	------	----------	-------------

PN sequence order for Sync Bits: 31

Number of Packets in Message: 44

Packet Repeat: 1

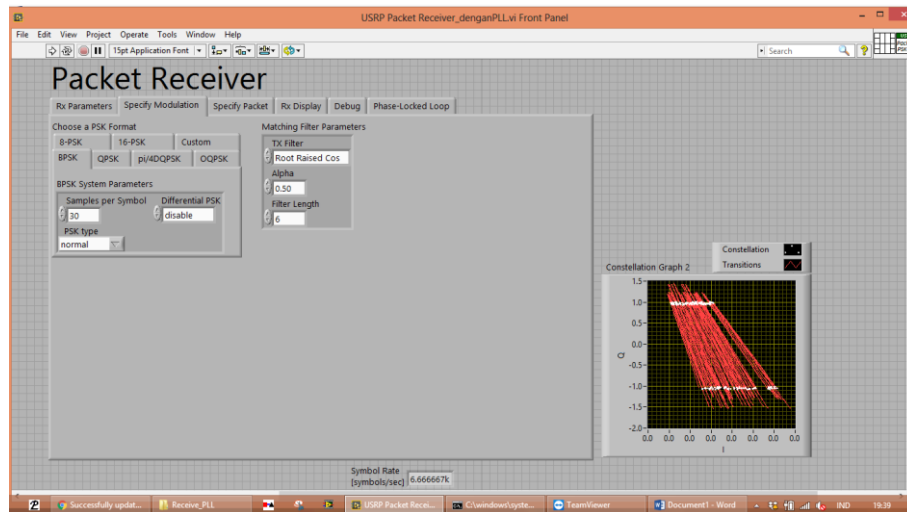
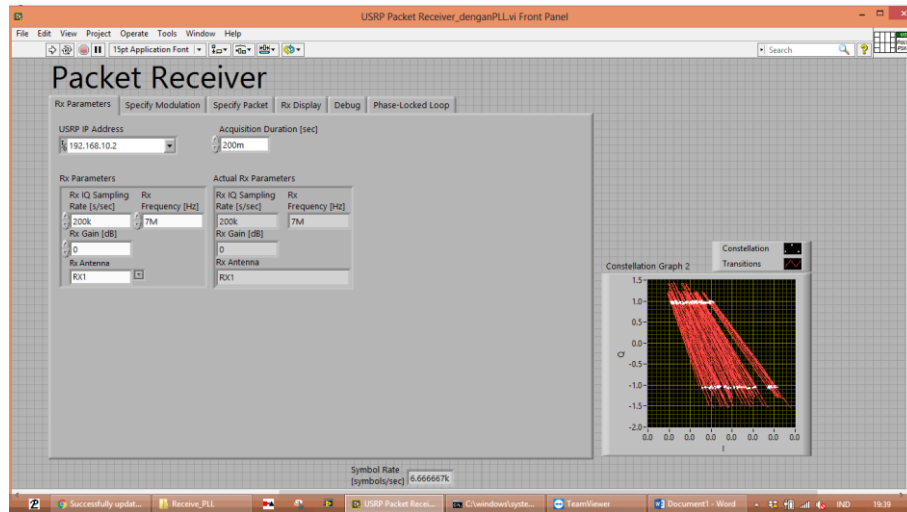
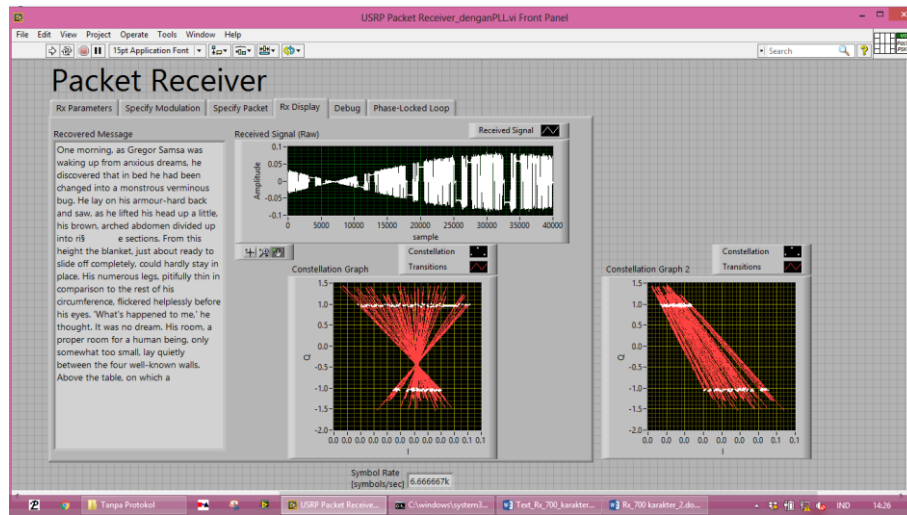
Constellation Graph

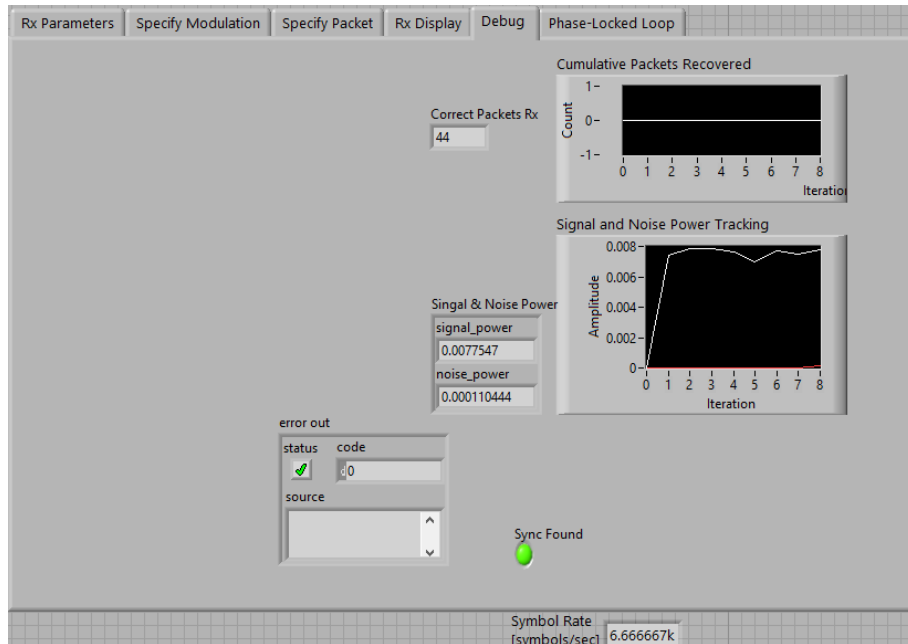
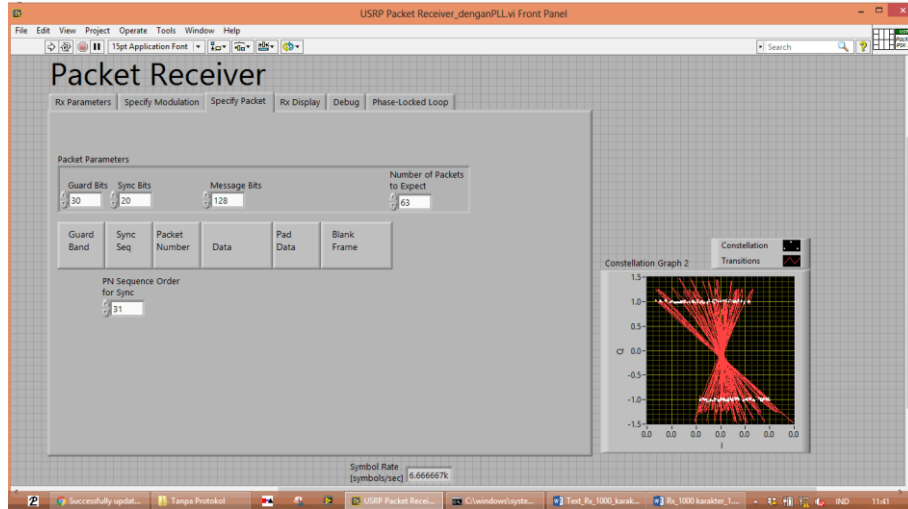
Autoscale

Symbol Rate [symbols/sec] 6,666667k

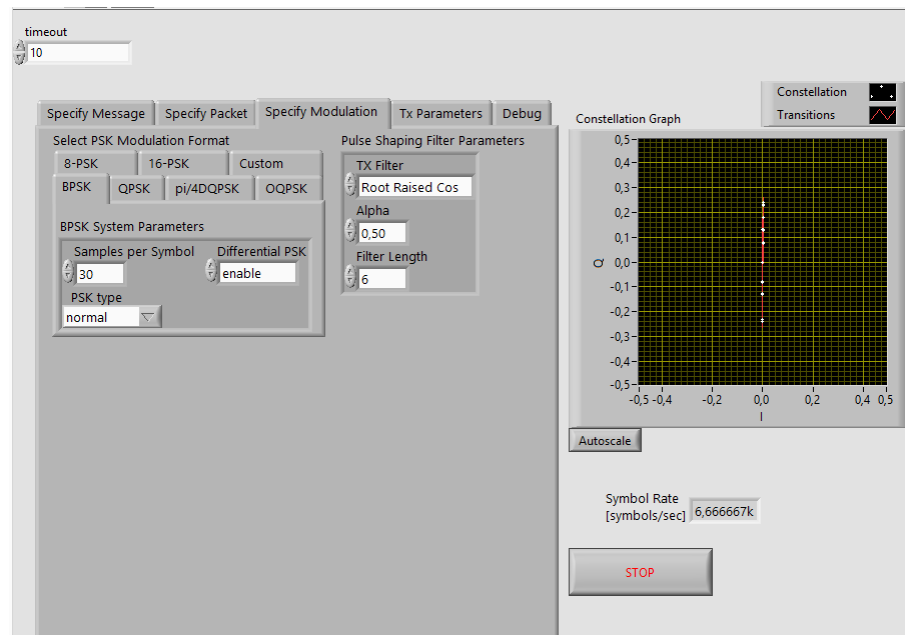
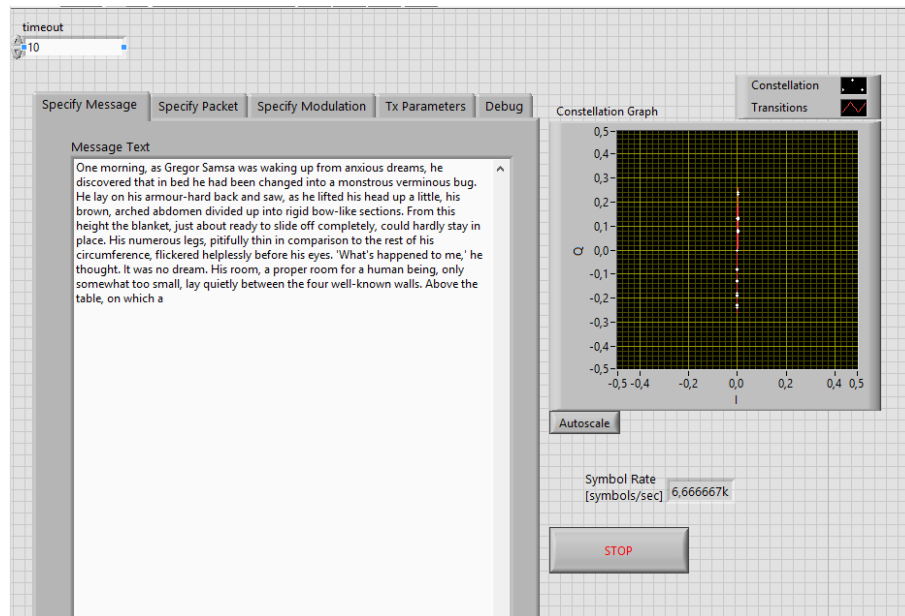
STOP

# Receiver :





## Dengan Protocol AX.25: Transmitter :



timeout  
10

Specify Message   Specify Packet   Specify Modulation   **Tx Parameters**   Debug

USRP IP address  
192.168.10.7

Tx Parameters		Actual Tx Parameters	
Tx IQ Sampling Rate [s/sec]	Tx Frequency [Hz]	Tx IQ Sampling Rate [s/sec]	Tx Frequency [Hz]
200k	7M	200k	7M
Tx Gain [dB]		Tx Gain [dB]	
0		0	
Tx Antenna		Tx Antenna	
TX1		TX1	

Constellation Graph  
Transitions

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

PSK system parameters

- samples per symbol (16): 30
- symbol map: 0, 6, 12, 30, 31
- differential PSK: enable
- PSK type: 0

Packet Parameters

- Guard Bits: 30
- Sync Bits: 20
- Message Bits: 128

Message Text: One morning.

TX Filter: Root

Pulse Shaping Filter Parameters

- Alpha: 0,5
- Filter Length: 6

Tx IQ Sampling Rate [s/sec]: 200000

Samples per Symbol: 30

Symbol Rate: 6666,67

Packets

Packet	t0	dt	Y
0	0	5E-6	0 + 0 i
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

message bitstream

1
1
1
0
1
1
1
1
1
1
1
1
1
1
0

Number of Packets: 44

passed ●

STR\_FROM\_CALLSIGN: YG3EGY    STR\_TO\_CALLSIGN: YG3GZ    CONTROL: 03    F0    FCS: 13115

text: One morning, as Gregor Samsa

message bitstream

23	0
	1
	1
	1
	1
	1
	1
	1
	0
	1
	1
	1

File Length (bits): 16000

Packet Message Length (bits): 128


Number of Packets: 44

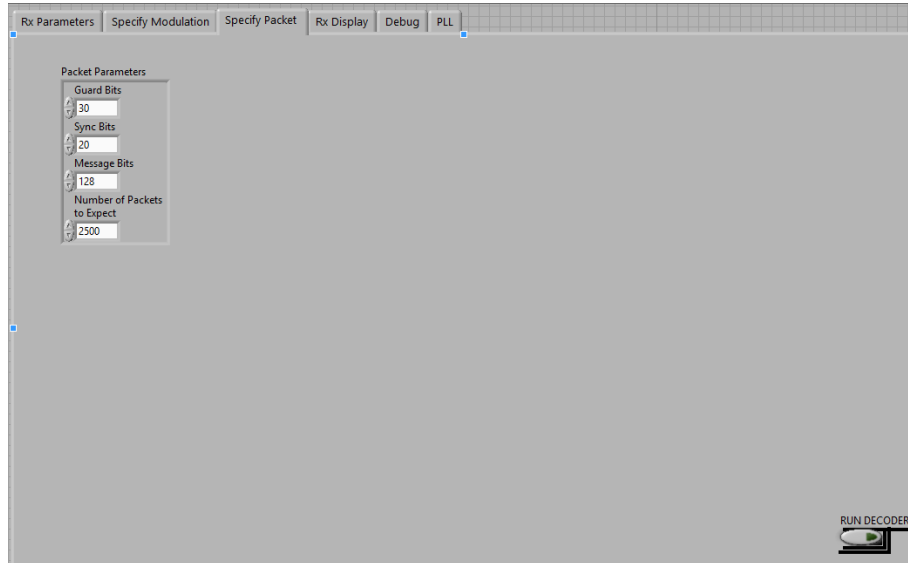
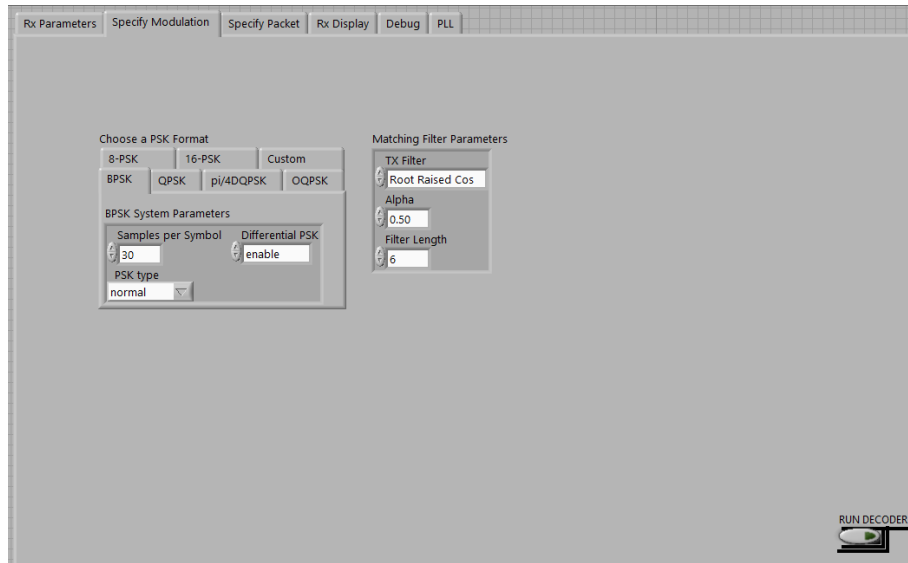
**Receiver :**

Rx Parameters    Specify Modulation    Specify Packet    Rx Display    Debug    PLL

USRP IP Address: 192.168.10.2    Acquisition Duration [sec]: 200m

Rx Parameters		Actual Rx Parameters	
Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]	Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]
200k	7M	200k	7M
Rx Gain [dB]: 0		Rx Gain [dB]: 0	
Rx Antenna: RX1		Rx Antenna: RX1	

RUN DECODER 





Checksum

0 255 255

0

Number of char

700

OUT\_MESSG

One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which a

Rx Parameters | Specify Modulation | Specify Packet | Rx Display | Debug | PLL

Cumulative Packets Recovered

Iteration	Count
0	1
1	0
2	0
3	-1

error out

status code

✓ 0

source

Sync Found

Correct Packets Rx

44

Signal & Noise Power

signal\_power

0.00769035

noise\_power

1.58959E-5

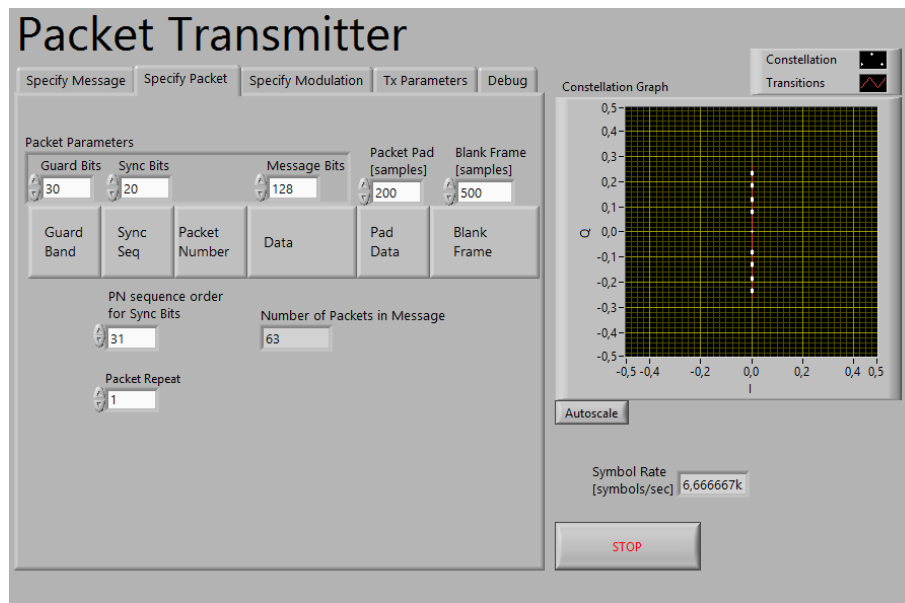
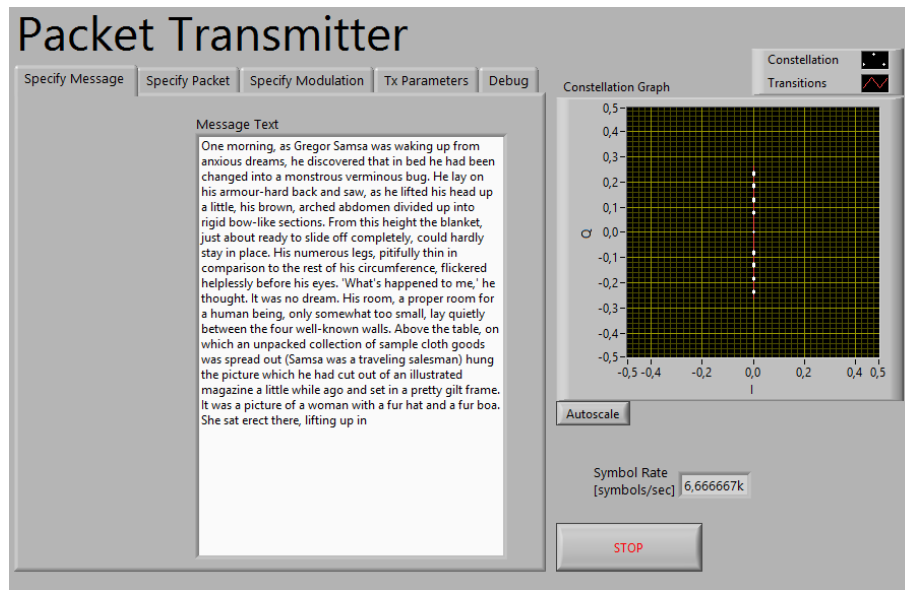
Signal and Noise Power Tracking

Iteration	Amplitude
0	0
1	0.006
2	0.006
3	0.006

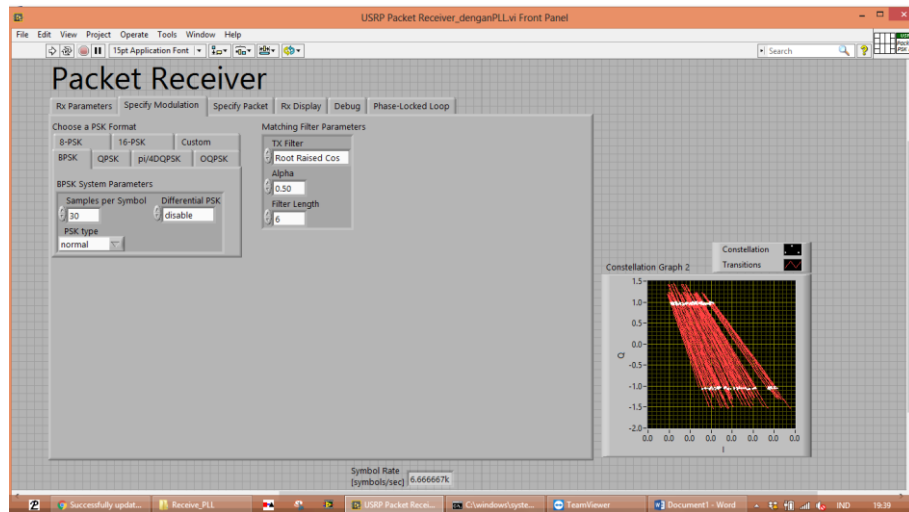
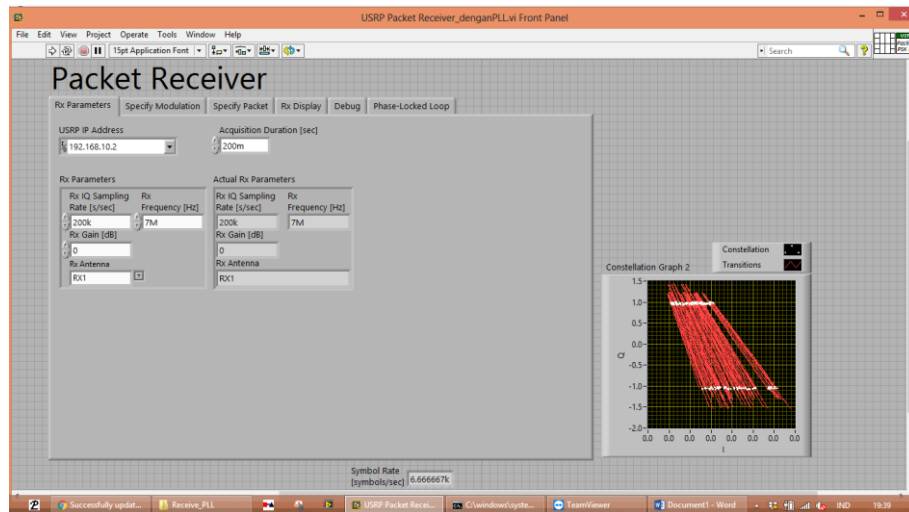
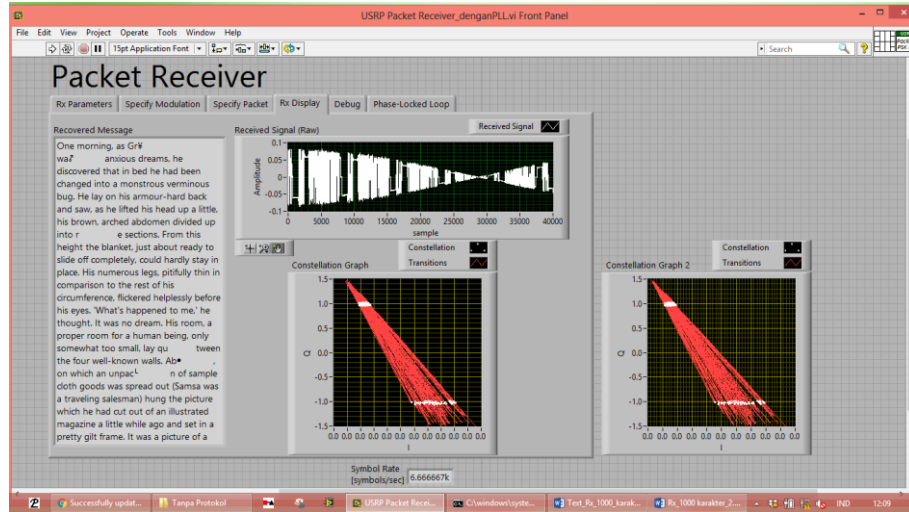
RUN DECODER

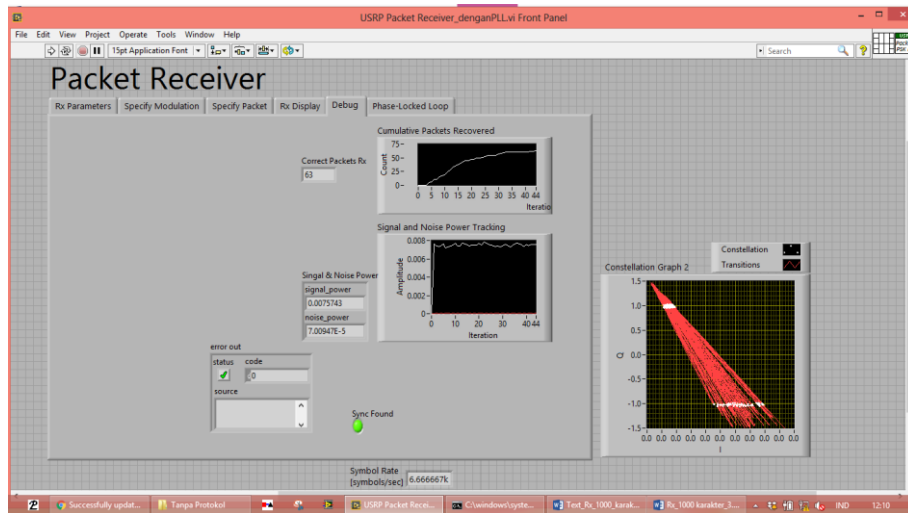
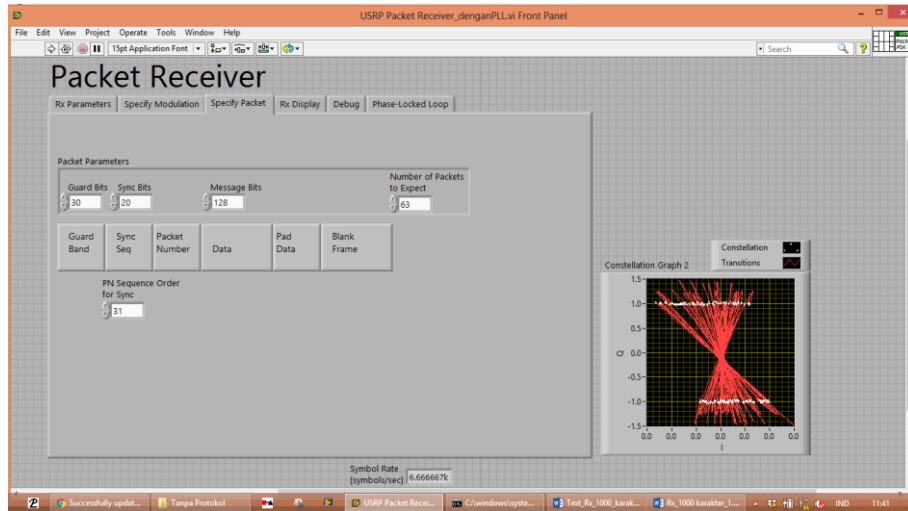
### 3. Contoh Pengujian Labview Tx dan Rx 1000 Karakter Tanpa Protocol AX.25 dan Dengan Protocol AX.25

#### Tanpa Protocol AX.25 : Transmitter :



## Receiver :





## Dengan Protocol AX.25: Transmitter :

The screenshot shows a software interface for a transmitter. At the top, there are tabs: "Specify Message", "Specify Packet", "Specify Modulation", "Tx Parameters", and "Debug". The "Specify Message" tab is active, displaying a text area with the following message:

Message Text  
One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which an unpacked collection of sample cloth goods was spread out (Samsa was a traveling salesman) hung the picture which he had cut out of an illustrated magazine a little while ago and set in a pretty gilt frame. It was a picture of a woman with a fur hat and a fur boa. She sat erect there, lifting up in

To the right of the text area is a "Constellation Graph" window. It shows a grid with axes ranging from -0.5 to 0.5. A vertical red dashed line is visible at the center (0,0). Above the graph, there are controls for "Constellation" and "Transitions". Below the graph is an "Autoscale" button. At the bottom right, there is a "Symbol Rate" field set to "6,666667k" [symbols/sec] and a red "STOP" button.

The screenshot shows the same software interface, but with the "Specify Modulation" tab active. It displays various modulation parameters:

Select PSK Modulation Format  
8-PSK | 16-PSK | Custom  
BPSK | QPSK | pi/4DQPSK | OQPSK

BPSK System Parameters  
Samples per Symbol: 30  
Differential PSK: enable  
PSK type: normal

Pulse Shaping Filter Parameters  
TX Filter: Root Raised Cos  
Alpha: 0,50  
Filter Length: 6

The "Constellation Graph" window is also visible, showing the same grid and red dashed line. The "Autoscale" button and "Symbol Rate" field (6,666667k) are also present. The "STOP" button is at the bottom right.

timeout  
10

Specify Message | Specify Packet | Specify Modulation | Tx Parameters | Debug

USRP IP address  
192.168.10.7

Tx Parameters

Tx IQ Sampling		Actual Tx Parameters	
Rate [s/sec]	Tx Frequency [Hz]	Rate [s/sec]	Tx Frequency [Hz]
200k	7M	200k	7M
Tx Gain [dB]		Tx Gain [dB]	
0		0	
Tx Antenna		Tx Antenna	
TX1		TX1	

Constellation Graph

Constellation Transitions

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

message bitstream

20

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

t0 0

dt 5E-6

9 0-0 i

Symbol Rate 6666,67

Number of Packets 63

PSK system parameters

samples per symbol (16) 30

symbol map 6,12303f

differential PSK enable 0

PSK type 0

Packet Parameters

Guard Bits 30

Sync Bits 20

Message Bits 128

Message Text One morning.

Tx IQ Sampling Rate [s/sec] 200000


Samples per Symbol 30

Pulse Shaping Filter Parameters

Alpha 0,5

Filter Length 6

TX Filter Root

passed 

STR_FROM_CALLSIGN	STR_TO_CALLSIGN	CONTROL	FCS
YG3EGY	YG3GZ	03	F0 13115

text  
One morning, as Gregor Samsa

message bitstream

23	0
	1
	1
	1
	1
	1
	1
	1
	0
	1
	1
	1

File Length (bits)  
16000

Packet Message Length (bits)  
128


Number of Packets  
63

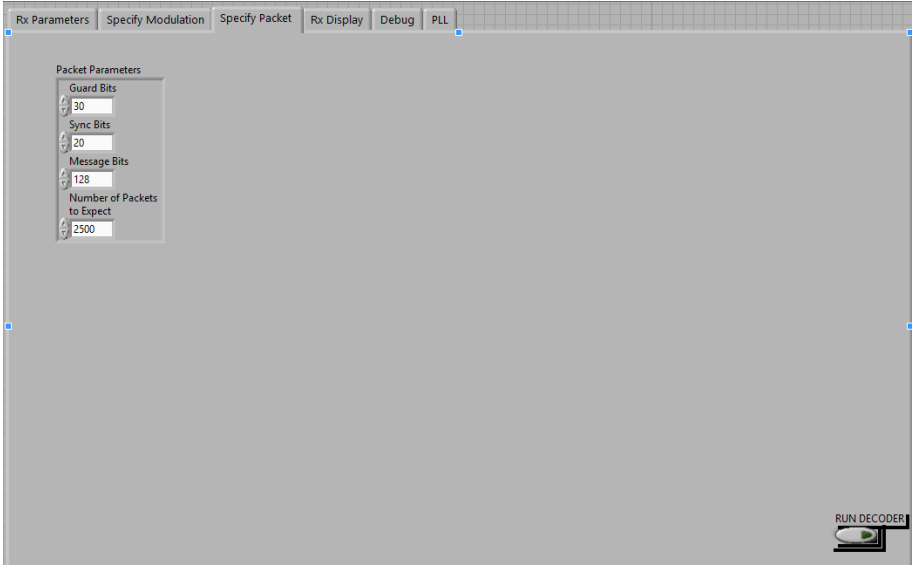
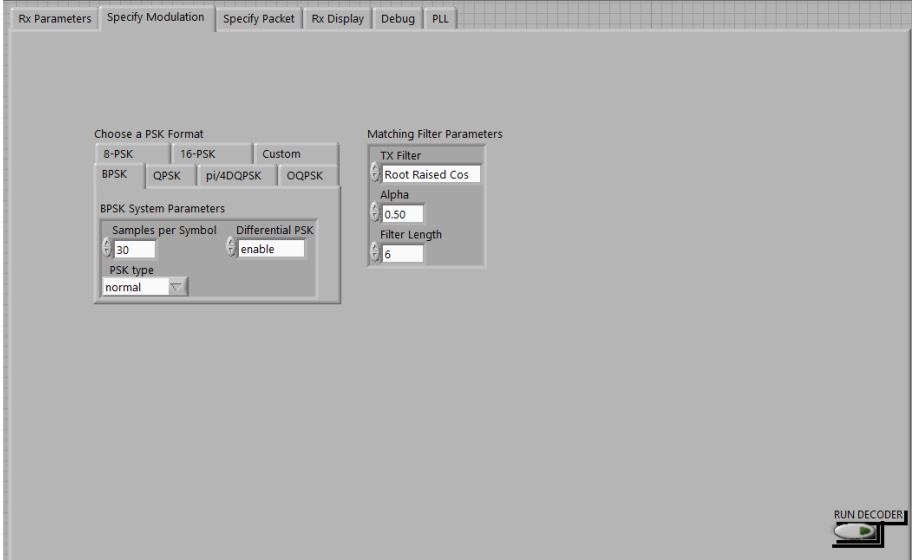
## Receiver :

Rx Parameters   Specify Modulation   Specify Packet   Rx Display   Debug   PLL

USRP IP Address: 192.168.10.2   Acquisition Duration [sec]: 200m

Rx Parameters		Actual Rx Parameters	
Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]	Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]
200k	7M	200k	7M
Rx Gain [dB]: 0		Rx Gain [dB]: 0	
Rx Antenna: RX1		Rx Antenna: RX1	







Checksum

0 255 255

0

Number of char

1000

OUT\_MESSG

One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which an unpacked collection of sample cloth goods was spread out (Samsa was a traveling salesman) hung the picture which he had cut out of an illustrated magazine a little while ago and set in a pretty gilt frame. It was a picture of a woman with a fur hat and a fur boa. She sat erect there, lifting up in

Rx Parameters Specify Modulation Specify Packet Rx Display Debug PLL

Cumulative Packets Recovered

Iteration	Count
0	1
1	0
2	-1

error out

status code

✓ 0

source

Sync Found

Correct Packets Rx

63

Signal & Noise Power

signal\_power

0.00734611

noise\_power

3.93861E-6

Signal and Noise Power Tracking

Iteration	Signal Power	Noise Power
0	0.0000	0.0000
1	0.0073	0.0000
2	0.0073	0.0000
3	0.0073	0.0000

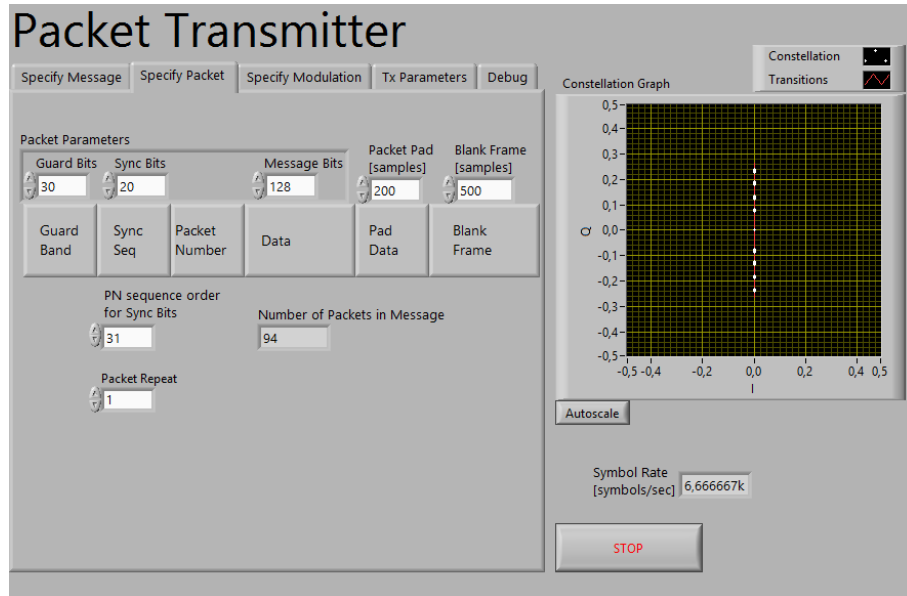
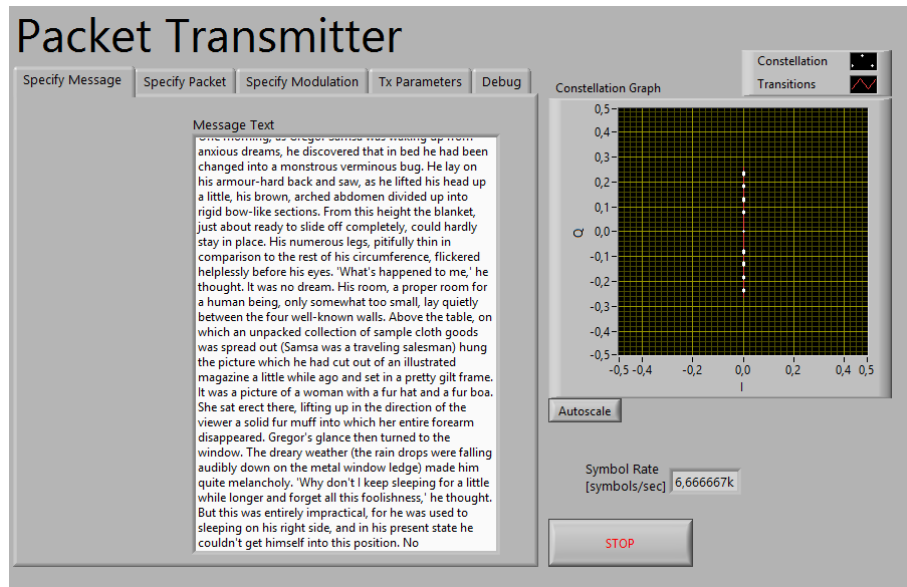
Amplitude

Iteration

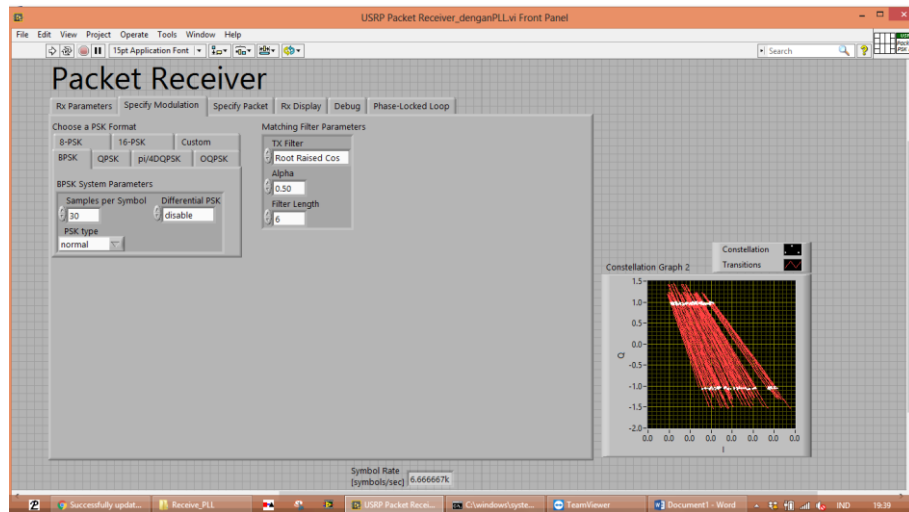
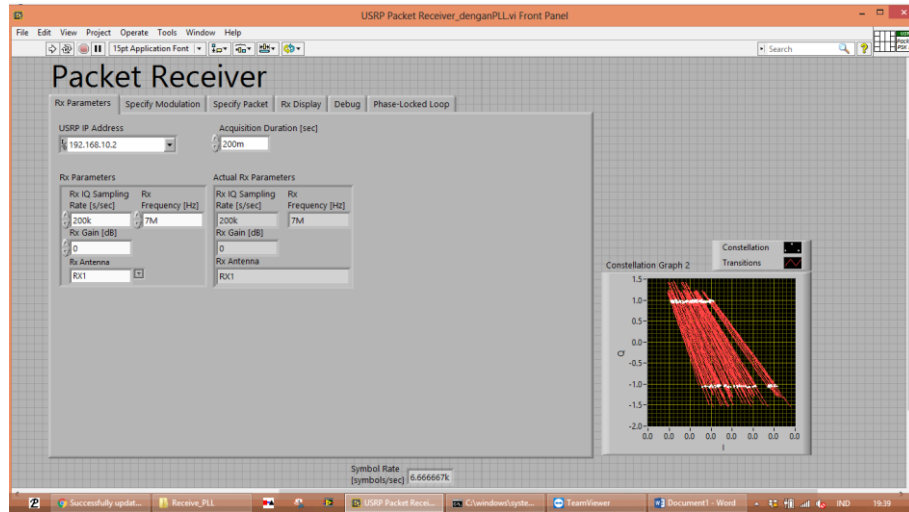
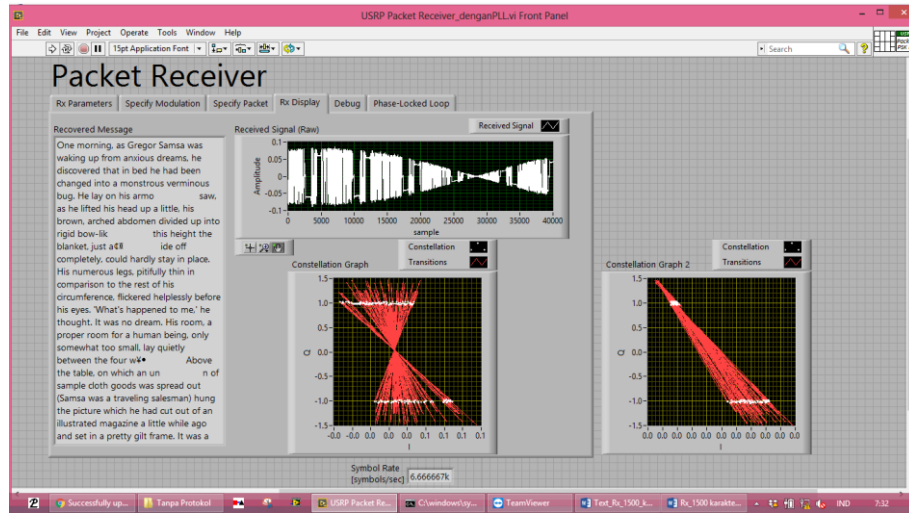
RUN DECODER

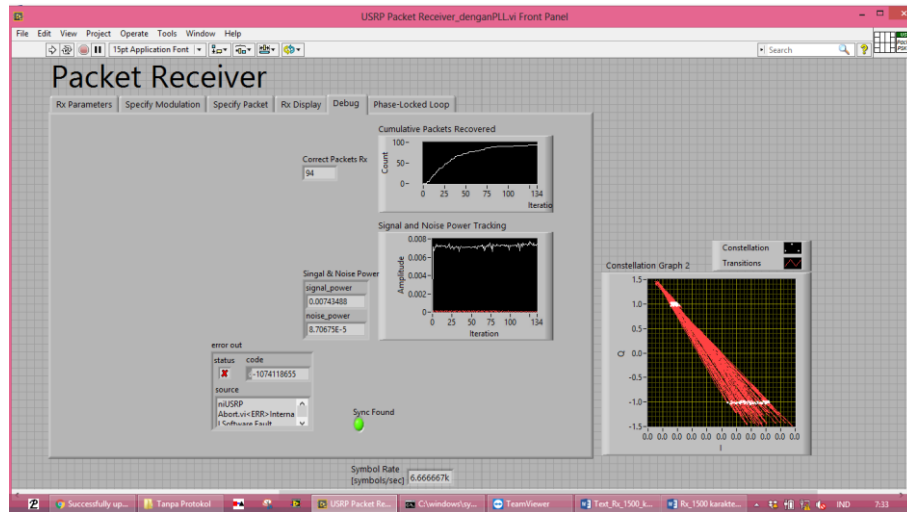
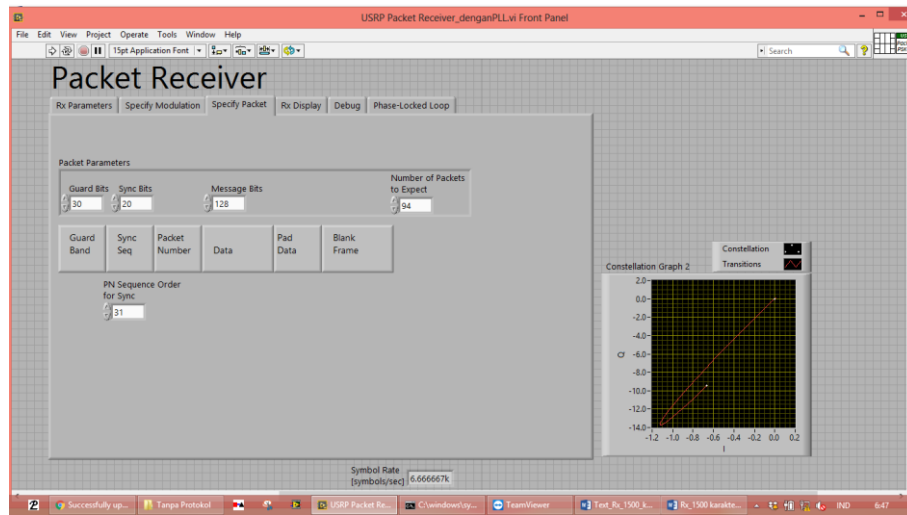
#### 4. Contoh Pengujian Labview Tx dan Rx 1500 Karakter Tanpa Protocol AX.25 dan Tanpa Protocol AX.25 :

Transmitter :



# Receiver :





## Dengan Protocol AX.25: Transmitter :

10

Specify Message | Specify Packet | Specify Modulation | Tx Parameters | Debug

Message Text

One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which an unpacked collection of sample cloth goods was spread out (Samsa was a traveling salesman) hung the picture which he had cut out of an illustrated magazine a little while ago and set in a pretty gilt frame. It was a picture of a woman with a fur hat and a fur boa. She sat erect there, lifting up in the direction of the viewer a solid fur muff into which her entire forearm disappeared. Gregor's glance then turned to the window. The dreary weather (the rain drops were falling audibly down on the metal window ledge) made him quite melancholy. 'Why don't I keep sleeping for a little while longer and forget all this foolishness,' he thought. But this was entirely impractical, for he was used to sleeping on his right side, and in his present state he couldn't get himself into this position. No

Constellation Graph

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

timeout

10

Specify Message | Specify Packet | Specify Modulation | Tx Parameters | Debug

Select PSK Modulation Format

8-PSK | 16-PSK | Custom

BPSK | QPSK | pi/4DQPSK | OQPSK

BPSK System Parameters

Samples per Symbol 30

Differential PSK enable

PSK type normal

Pulse Shaping Filter Parameters

TX Filter Root Raised Cos

Alpha 0,50

Filter Length 6

Constellation Graph

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

timeout  
10

Specify Message | Specify Packet | Specify Modulation | Tx Parameters | Debug

USRP IP address  
192.168.10.7

**Tx Parameters**

Tx IQ Sampling Rate [s/sec]	Tx Frequency [Hz]	Tx Gain [dB]	Tx Antenna
200k	7M	0	TX1

**Actual Tx Parameters**

Tx IQ Sampling Rate [s/sec]	Tx Frequency [Hz]	Tx Gain [dB]	Tx Antenna
200k	7M	0	TX1

Constellation Graph

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

SK system parameters

samples per symbol (16) 30

symbol map 6,12303i

differential PSK enable 0

PSK type 0

Message Text One morning.

Packet Parameters

Guard Bits 30

Sync Bits 20

Message Bits 128

Samples per Symbol 30

Tx IQ Sampling Rate [s/sec] 200000

Pulse Shaping Filter Parameters

Alpha 0,5

Filter Length 6

TX Filter Root

Packets 20

Symbol Rate 6666,67

message bitstream

1
1
1
0
1
1
1
1
1
1
1
1
1
0

Number of Packets 94

passed ●

STR_FROM_CALLSIGN	STR_TO_CALLSIGN	CONTROL	FCS
YG3EGY	YG3GZ	03	F0 13115

text One morning, as Gregor Samsa	message bitstream 23 0 1 1 1 1 1 1 1 0 1 1 1	File Length (bits) 16000 Packet Message Length (bits) 128 Number of Packets 94
--------------------------------------	---	---

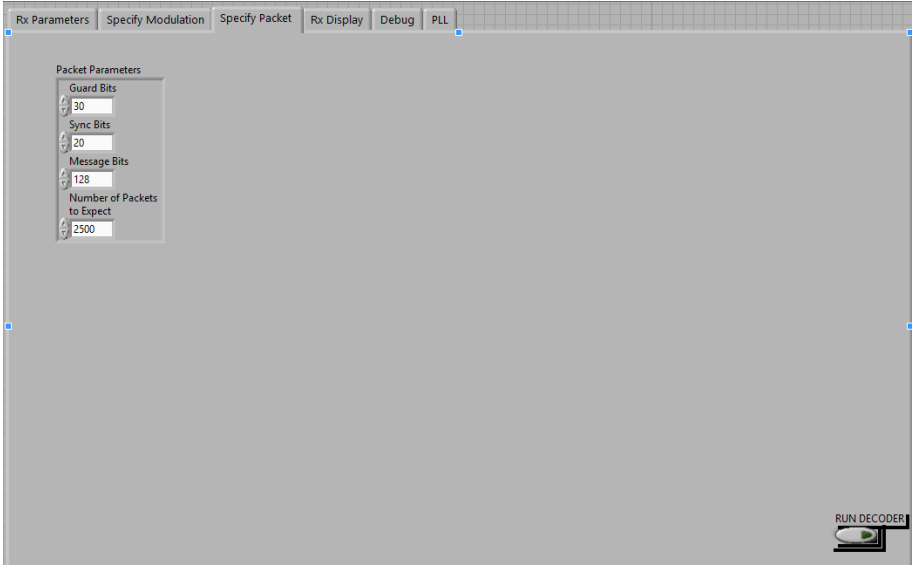
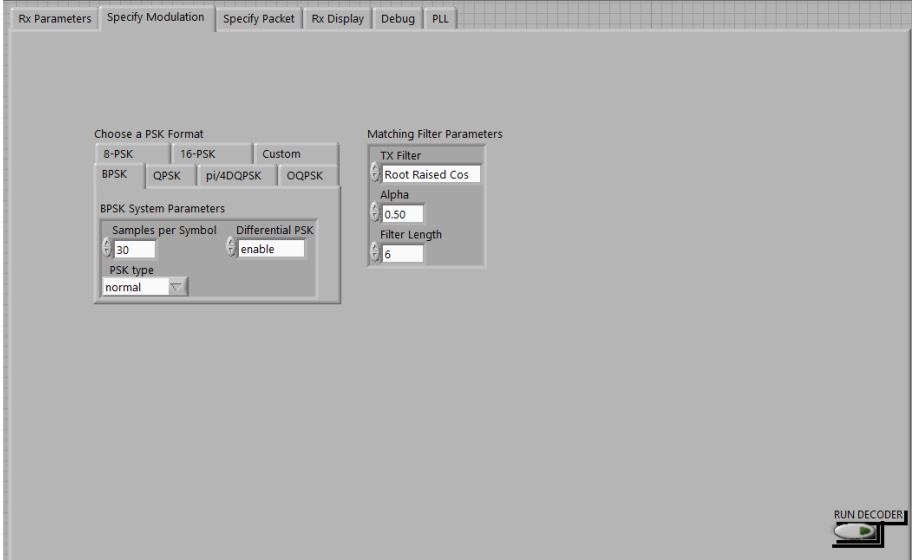
**Receiver :**

Rx Parameters | Specify Modulation | Specify Packet | Rx Display | Debug | PLL

USRP IP Address 192.168.10.2	Acquisition Duration [sec] 200m
---------------------------------	------------------------------------

Rx Parameters		Actual Rx Parameters	
Rx IQ Sampling Rate [s/sec] 200k	Rx Frequency [Hz] 7M	Rx IQ Sampling Rate [s/sec] 200k	Rx Frequency [Hz] 7M
Rx Gain [dB] 0		Rx Gain [dB] 0	
Rx Antenna RX1		Rx Antenna RX1	

RUN DECODER





Checksum

0 255 255

0

Number of char

1500

OUT\_MESSG

One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted his head up a little, his brown, arched abdomen divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which an unpacked collection of sample cloth goods was spread out (Samsa was a traveling salesman) hung the picture which he had cut out of an illustrated magazine a little while ago and set in a pretty gilt frame. It was a picture of a woman with a fur hat and a fur boa. She sat erect there, lifting up in the direction of the viewer a solid fur muff into which her entire forearm disappeared. Gregor's glance then turned to the window. The dreary weather (the rain drops were falling audibly down on the metal window ledge) made him quite melancholy. 'Why don't I keep sleeping for a little while longer and forget all this foolishness,' he thought. But this was entirely impractical, for he was used to sleeping on his right side, and in his present state he couldn't get himself into this position.

No

Rx Parameters   Specify Modulation   Specify Packet   Rx Display   Debug   PLL

Cumulative Packets Recovered

Iteration	Count
0	0
1	1
2	2
3	3
4	4
5	5

error out

status code  0

source

Sync Found ●

Correct Packets Rx

Signal & Noise Power

signal\_power

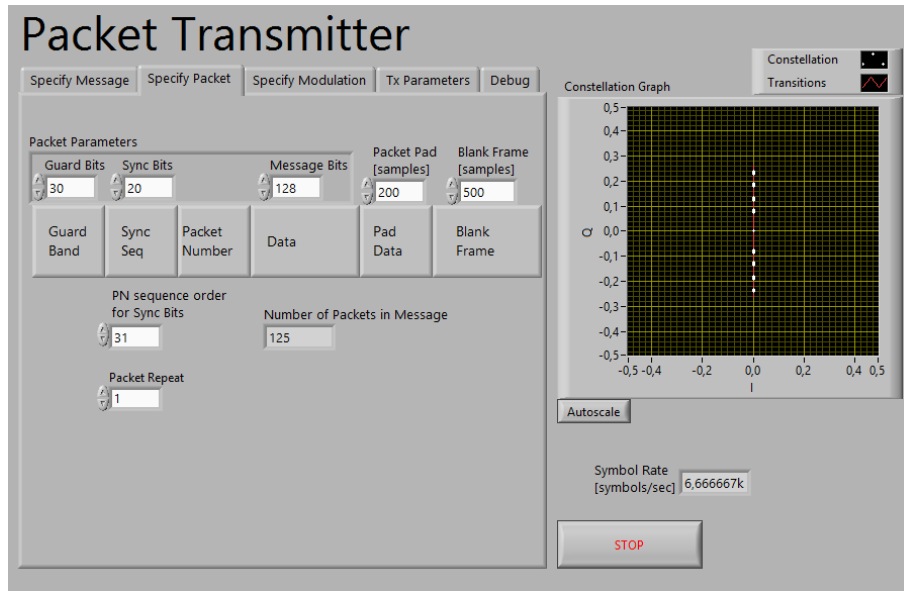
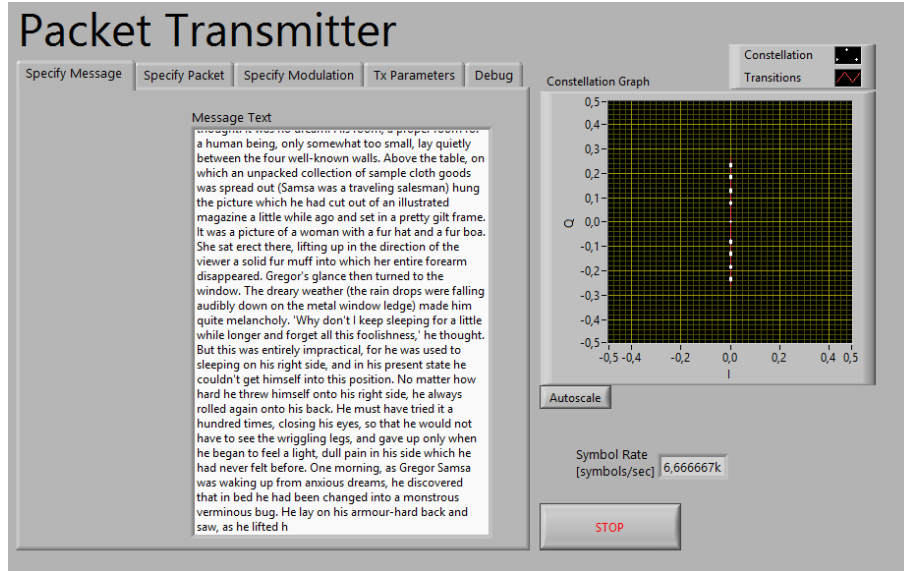
noise\_power

Signal and Noise Power Tracking

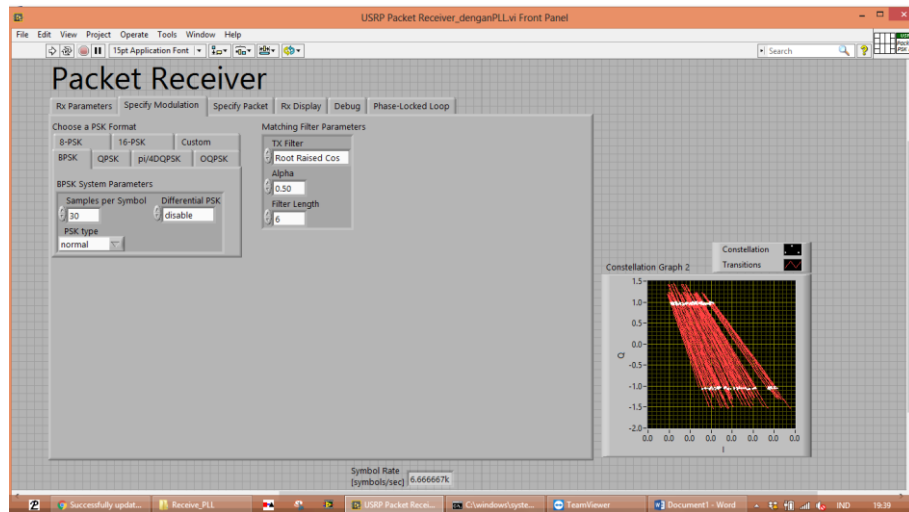
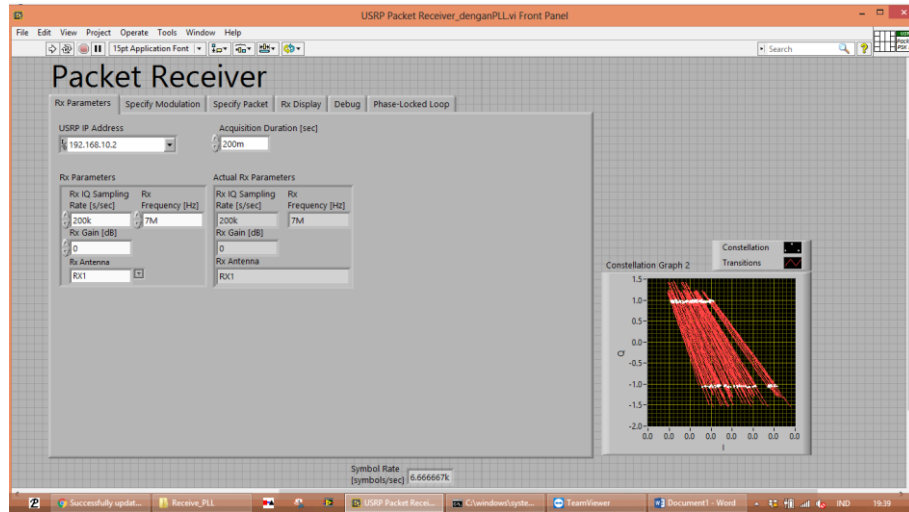
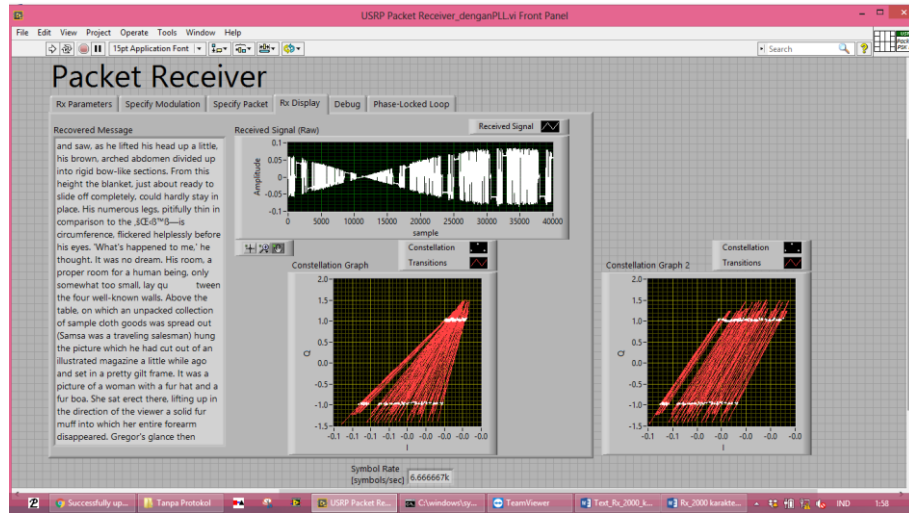
Iteration	Amplitude
0	0
1	0.00745066
2	0.00745066
3	0.00745066
4	0.00745066
5	0.00745066

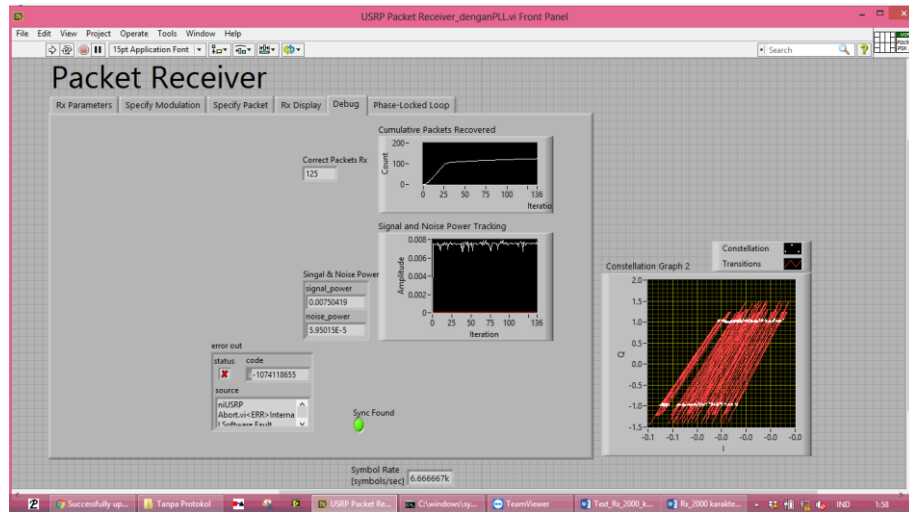
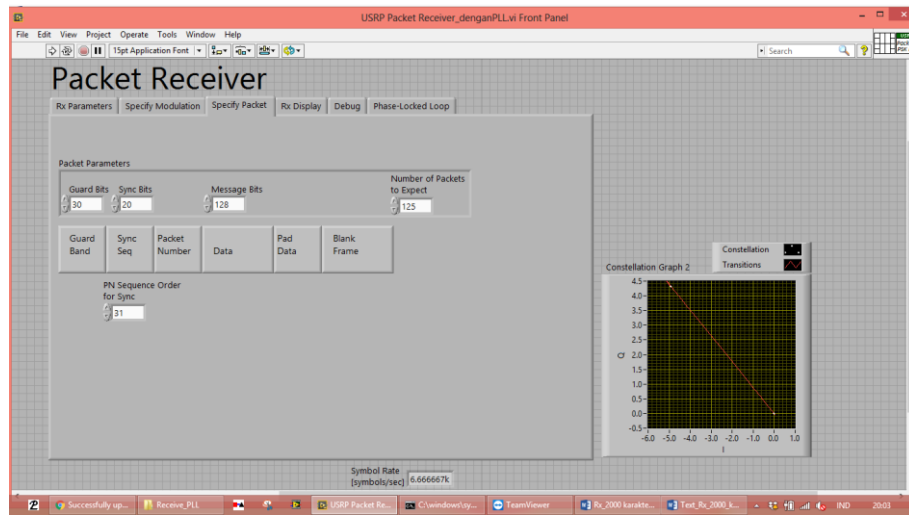
RUN DECODER

**5. Contoh Pengujian Labview Tx dan Rx 2000 Karakter Tanpa Protocol AX.25 dan Dengan Protocol AX.25 :  
Tanpa Protocol AX.25 :  
**Transmitter :****



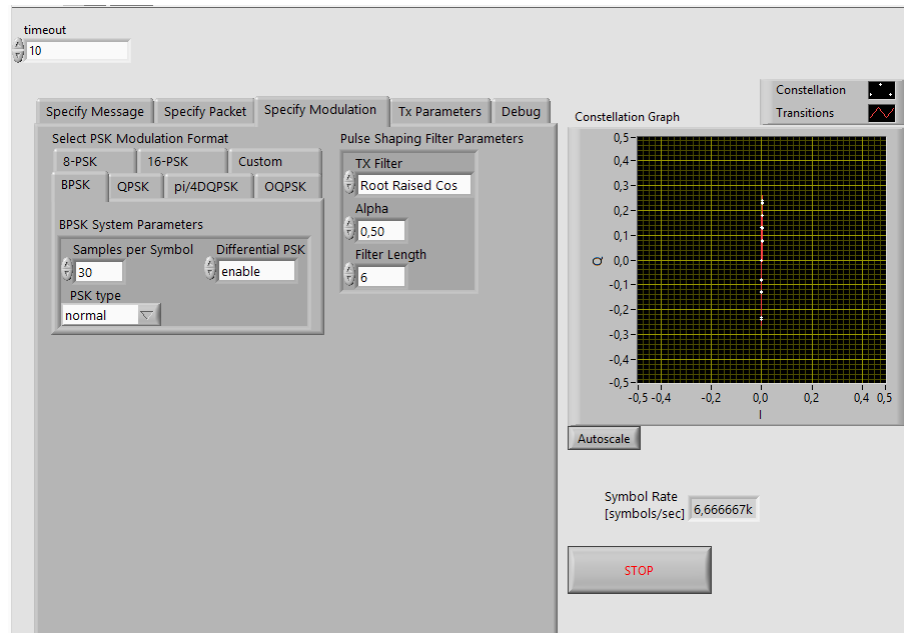
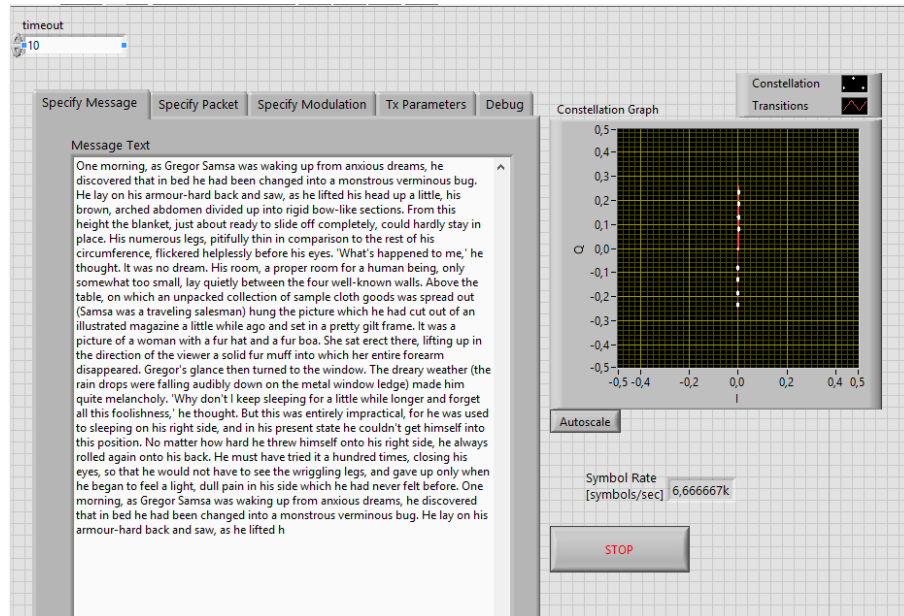
# Receiver :





## Dengan Protocol AX.25:

### Transmitter :



timeout  
10

Specify Message   Specify Packet   Specify Modulation   Tx Parameters   Debug

USRP IP address  
192.168.10.7

Tx Parameters   Actual Tx Parameters

Tx IQ Sampling Rate [s/sec]	Tx Frequency [Hz]	Tx IQ Sampling Rate [s/sec]	Tx Frequency [Hz]
200k	7M	200k	7M
Tx Gain [dB]		Tx Gain [dB]	
0		0	
Tx Antenna		Tx Antenna	
TX1		TX1	

Constellation Graph

Constellation Transitions

Autoscale

Symbol Rate [symbols/sec] 6,666667k

STOP

PSK system parameters

samples per symbol (16) 30

symbol map 6,123031

differential PSK enable

PSK type 0

Packet Parameters

Guard Bits 30

Sync Bits 20

Message Bits 128

Message Text One morning.

Samples per Symbol 30

Tx IQ Sampling Rate [s/sec] 200000

Pulse Shaping Filter Parameters

Alpha 0.5

Filter Length 6

TX Filter Root

Packets 20

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

message bitstream

Symbol Rate 6666,67

Number of Packets 125

passed ●

STR_FROM_CALLSIGN	STR_TO_CALLSIGN	CONTROL	FCS
YG3EGY	YG3GZ	03	F0 13115

text	message bitstream																								
One morning, as Gregor Samsa	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>23</td><td>0</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>0</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> <tr><td></td><td>1</td></tr> </table>	23	0		1		1		1		1		1		1		1		0		1		1		1
23	0																								
	1																								
	1																								
	1																								
	1																								
	1																								
	1																								
	1																								
	0																								
	1																								
	1																								
	1																								

File Length (bits)	16000
Packet Message Length (bits)	128
Number of Packets	125

## Receiver :

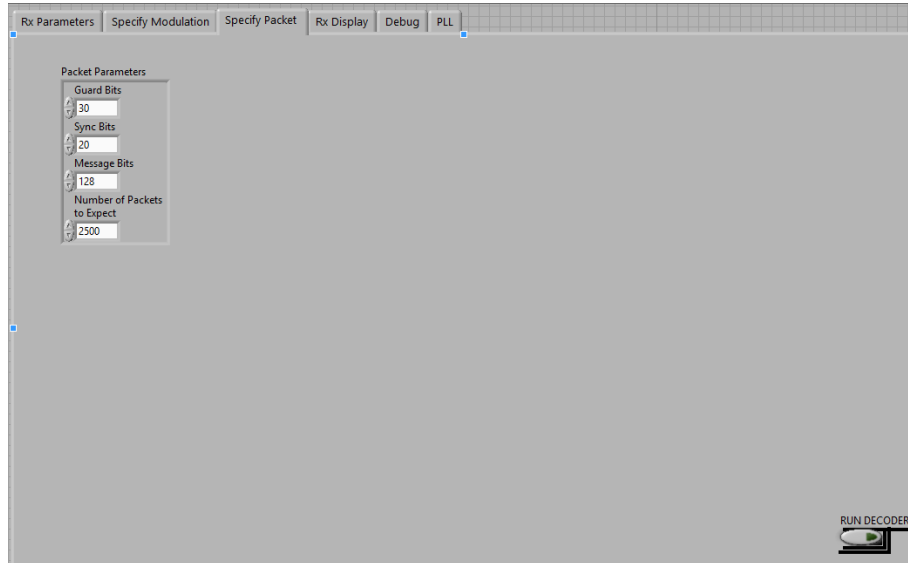
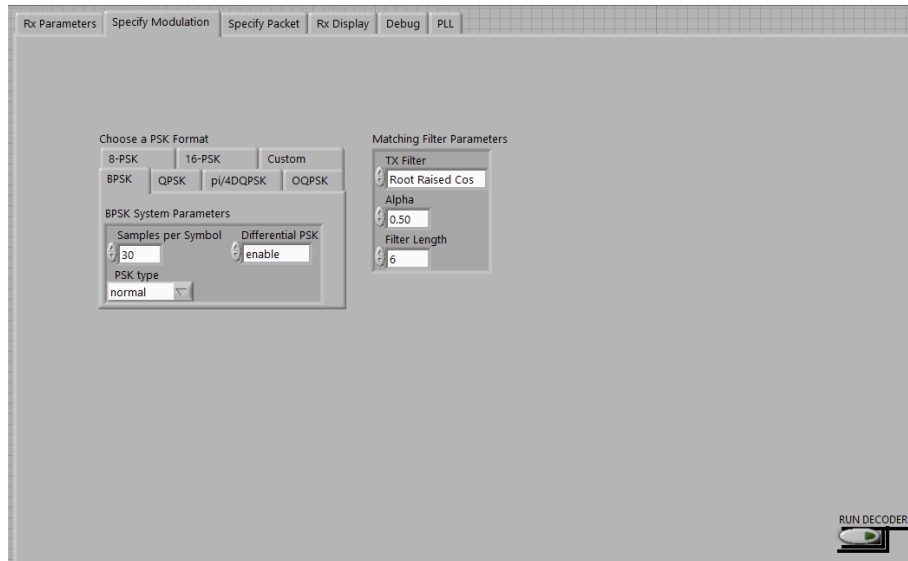
Rx Parameters    Specify Modulation    Specify Packet    Rx Display    Debug    PLL

USRP IP Address	Acquisition Duration [sec]
192.168.10.2	200m

Rx Parameters		Actual Rx Parameters	
Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]	Rx IQ Sampling Rate [s/sec]	Rx Frequency [Hz]
200k	7M	200k	7M
Rx Gain [dB]		Rx Gain [dB]	
0		0	
Rx Antenna		Rx Antenna	
RX1		RX1	

RUN DECODER





Checksum

0 255 255

0

Number of char

2000

OUT\_MESSG

divided up into rigid bow-like sections. From this height the blanket, just about ready to slide off completely, could hardly stay in place. His numerous legs, pitifully thin in comparison to the rest of his circumference, flickered helplessly before his eyes. 'What's happened to me,' he thought. It was no dream. His room, a proper room for a human being, only somewhat too small, lay quietly between the four well-known walls. Above the table, on which an unpacked collection of sample cloth goods was spread out (Samsa was a traveling salesman) hung the picture which he had cut out of an illustrated magazine a little while ago and set in a pretty gilt frame. It was a picture of a woman with a fur hat and a fur boa. She sat erect there, lifting up in the direction of the viewer a solid fur muff into which her entire forearm disappeared. Gregor's glance then turned to the window. The dreary weather (the rain drops were falling audibly down on the metal window ledge) made him quite melancholy. 'Why don't I keep sleeping for a little while longer and forget all this foolishness,' he thought. But this was entirely impractical, for he was used to sleeping on his right side, and in his present state he couldn't get himself into this position. No matter how hard he threw himself onto his right side, he always rolled again onto his back. He must have tried it a hundred times, closing his eyes, so that he would not have to see the wriggling legs, and gave up only when he began to feel a light, dull pain in his side which he had never felt before. One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in bed he had been changed into a monstrous verminous bug. He lay on his armour-hard back and saw, as he lifted h

Rx Parameters   Specify Modulation   Specify Packet   Rx Display   Debug   PLL

Cumulative Packets Recovered

error out

status	code
✓	0

source

Sync Found

Correct Packets Rx: 125

Signal & Noise Power

signal_power	0.00715116
noise_power	0.000140171

Signal and Noise Power Tracking

RUN DECODER

*Halaman ini sengaja dikosongkan*

## LAMPIRAN 2

### Listing Program Protocol AX.25 di Tx dan Rx

#### ENCODER

```
%bitsequence = zeros(5,1);
%=====
AX25_MAX_PAYLOAD_SIZE = 160;
CRC_POLYGEN = hex2dec('1021');
MAX_LENGTH = 280;
MAX_LENGTH_FINAL = 450+256+256+256+256+256+256;
BYTE_NUM = 8;
MSG_LENGTH = 256;
%=====
bitsequence =zeros(size(MAX_LENGTH*BYTE_NUM));
finalSequence =zeros(size(MAX_LENGTH_FINAL));
RcvSequence = zeros(size(MAX_LENGTH_FINAL));
messages = [];%zeros(size(MSG_LENGTH));
%=====
STR_FROM_CALLSIGN = 'YG3EGY';
STR_TO_CALLSIGN = 'A002';
CONTROL = '03';%'0x03';
PID = 'F0';%'0xF0';
%=====
SrcCallsign=['Y','G','3','E','G','0','Y'];%zeros(size(7));
DestCallsign=['A','0','0','0','0','0','2'];%zeros(size(7));
ssid_source = 0;
ssid_destination = 0;
%=====
%FCS = uint64(0);
c =1;
for i=1:1:(MAX_LENGTH*BYTE_NUM)
    bitsequence(c)=0;
    c=c+1;
end
c =1;
for i=1:1:(MAX_LENGTH*BYTE_NUM)
    bitsequence(c)=0;
    c=c+1;
end
c=1;
for i=1:1:(MAX_LENGTH_FINAL)
    finalSequence(c) = 0;
    RcvSequence(c) = 0;
    c=c+1;
end;
c=1;
```

```

for i=1:1:(MSG_LENGTH)
    messages(c) = hex2dec('00');%white space
    c=c+1;
end;
%=====
% TEST MESSAGES
%INPUT_MSSG = 'TESTING AX';
txtToSending= reshape(INPUT_MSSG, 1, []);
%str copy to messages
c =1;
for i=1:1: length(txtToSending)
    messages(c) = txtToSending(1,c);
    c = c+1;
end
%=====
%ADD HEADER
%Shift bits 1 place to the left in order to allow for HDLC
extension bit
%=====
c =1;
Index =1;
for i=1:1:length(DestCallsign)
    %Append SSID Destination
    bitsequence(Index) = bitshift((uint16(DestCallsign(i))),1);
    c = c+1;
    Index = Index+1;
end
bitsequence(Index) = hex2dec('60');%ssid_destination;
Index = Index+1;
%=====
%Append Source Callsign
%=====
c =1;
for i=1:1:length(SrcCallsign)
    bitsequence(Index) = bitshift(uint16(SrcCallsign(i)),1);
    %fprintf(num2str(bitsequence(Index)));
    %fprintf(' ');
    Index = Index+1;
    c =c+1;
end
bitsequence(Index) = hex2dec('61');%ssid_source;
Index = Index+1;
%=====
%Append Control bits
bitsequence(Index) = hex2dec(CONTROL);
Index = Index+1;
%Append Protocol Identifier

```

```

bitsequence(Index) = hex2dec(PID);
Index = Index+1;
%=====
    %Add Message
c =1;
for i=1:1: length(txtToSending)
    bitsequence(Index) = txtToSending(1,c);
    Index = Index+1;
    c = c+1;
end
%*****
%*****
%=====
%Convert bit sequence from MSB to LSB
%for (int i=0; i < Index ; i++) bitSequence[i] =
%MSB_LSB_swap_8bit(bitSequence[i]);
v =0;
c =1;
p =0;
    for i=1:1:Index-1
        v = (bitsequence(i));
        bitsequence(i) = MSB_LSB_swap_8bit(v);
    end
    %AFTER MSB LSB SWAP
    %=====
    FCS =
13115;%typecast(int16(13115),'uint16');%bin2dec('11001100111011'
);%13115;%CRC_CCITT(bitsequence,Index-1);
fprintf('FCS:');
fprintf(num2str(FCS));
fprintf('\n');
    %Add FCS in MSB form
%Add MS byte
%bitSequence[Index++] = (FCS >> 8) & 0xff;
%fprintf('FCS:');
%fprintf(num2str(FCS));
%fprintf('\n');
p = bitand(bitshift(FCS,-8),hex2dec('FF'));
    fprintf(':');
fprintf(num2str(p));
fprintf('\n');
    bitsequence(Index) =p;% bin2dec(num2str(p));
%fprintf(num2str(p));
%fprintf('\n');
Index = Index + 1;
%Add LS byte
%bitSequence[Index++] = FCS & 0xff;

```

```

p = bitand(FCS,hex2dec('FF'));
fprintf(num2str(p));
fprintf('\n');
bitsequence(Index) = p;%bin2dec(num2str(p));
Index = Index + 1;
=====
fprintf('Length : ');
for i=1:1:Index-1
    fprintf(num2str(bitsequence(i)));
    fprintf(' ');
end
fprintf('\n');
%FINAL RESULT ENCODER
%FINAL_RESULT = (1:450);
[ENCODE,indx] = BIT_PROCESSING(bitsequence,Index-
1,finalSequence);
for i=1:1:450+256+256+256+256+256+256+256
    FINAL_RESULT(i)=char(ENCODE(i));
end

```

## **DECODER**

```

%bitsequence = zeros(5,1);
=====
AX25_MAX_PAYLOAD_SIZE = 160;
CRC_POLYGEN = hex2dec('1021');
MAX_LENGTH = 280;
MAX_LENGTH_FINAL = 450+256+256+256+256+256+256+256;
BYTE_NUM = 8;
MSG_LENGTH = 256;
=====
bitsequence =zeros(size(MAX_LENGTH*BYTE_NUM));
finalSequence =zeros(size(MAX_LENGTH_FINAL));
RcvSequence = zeros(size(MAX_LENGTH_FINAL));
messages = [];%zeros(size(MSG_LENGTH));
=====
STR_FROM_CALLSIGN = 'YG3EGY';
STR_TO_CALLSIGN = 'A002';
CONTROL = '03';%'0x03';
PID = 'F0';%'0xF0';
=====
SrcCallsign=['Y','G','3','E','G','0','Y'];%zeros(size(7));
DestCallsign=['A','0','0','0','0','0','2'];%zeros(size(7));
ssid_source = 0;
ssid_destination = 0;
=====
%FCS = uint64(0);

```

```

c =1;
for i=1:1:(MAX_LENGTH*BYTE_NUM)
    bitsequence(c)=0;
    c=c+1;
end
c =1;
for i=1:1:(MAX_LENGTH*BYTE_NUM)
    bitsequence(c)=0;
    c=c+1;
end
c=1;
for i=1:1:(MAX_LENGTH_FINAL)
    finalSequence(c) = 0;
    RcvSequence(c) = 0;
    c=c+1;
end;
c=1;
for i=1:1:(MSG_LENGTH)
    messages(c) = hex2dec('00');%white space
    c=c+1;
end;
%=====
% TEST MESSAGES
%INPUT_MSSG = 'TESTING AX';
txtToSending= reshape(INPUT_MSSG, 1, []);
%str copy to messages
c =1;
for i=1:1: length(txtToSending)
    messages(c) = txtToSending(1,c);
    c = c+1;
end
%=====
%ADD HEADER
%Shift bits 1 place to the left in order to allow for HDLC
extension bit
%=====
c =1;
Index =1;
for i=1:1:length(DestCallsign)
    %Append SSID Destination
    bitsequence(Index) = bitshift((uint16(DestCallsign(i))),1);
    c = c+1;
    Index = Index+1;
end
bitsequence(Index) = hex2dec('60');%ssid_destination;
Index = Index+1;
%=====

```

```

%Append Source Callsign
%=====
c =1;
for i=1:1:length(SrcCallsign)
    bitsequence(Index) = bitshift(uint16(SrcCallsign(i)),1);
    %fprintf(num2str(bitsequence(Index)));
    %fprintf(' ');
    Index = Index+1;
    c =c+1;
end
bitsequence(Index) = hex2dec('61');%ssid_source;
Index = Index+1;
%=====
%Append Control bits
bitsequence(Index) = hex2dec(CONTROL);
Index = Index+1;
%Append Protocol Identifier
bitsequence(Index) = hex2dec(PID);
Index = Index+1;
%=====
%Add Message
c =1;
for i=1:1: length(txtToSending)
    bitsequence(Index) = txtToSending(1,c);
    Index = Index+1;
    c = c+1;
end
%*****
%*****
%=====
%Convert bit sequence from MSB to LSB
%for (int i=0; i < Index ; i++) bitSequence[i] =
%MSB_LSB_swap_8bit(bitSequence[i]);
v =0;
c =1;
p =0;
for i=1:1:Index-1
    v = (bitsequence(i));
    bitsequence(i) = MSB_LSB_swap_8bit(v);
end
%AFTER MSB LSB SWAP
%=====
FCS =
13115;%typecast(int16(13115),'uint16');%bin2dec('11001100111011'
);%13115;%CRC_CCITT(bitsequence,Index-1);
fprintf('FCS:');
fprintf(num2str(FCS));

```



```

fprintf('\n');

%Add FCS in MSB form
%Add MS byte
%bitSequence[Index++] = (FCS >> 8) & 0xff;
%fprintf('FCS:');
%fprintf(num2str(FCS));
%fprintf('\n');
p = bitand(bitshift(FCS,-8),hex2dec('FF'));
fprintf(':');
fprintf(num2str(p));
fprintf('\n');

bitsequence(Index) =p;% bin2dec(num2str(p));
%fprintf(num2str(p));
%fprintf('\n');
Index = Index + 1;
%Add LS byte
%bitSequence[Index++] = FCS & 0xff;
p = bitand(FCS,hex2dec('FF'));
%fprintf(num2str(p));
%fprintf('\n');
bitsequence(Index) = p;%bin2dec(num2str(p));
Index = Index + 1;
%=====
fprintf('Length : ');
for i=1:1:Index-1
    fprintf(num2str(bitsequence(i)));
    fprintf(' ');
end
fprintf('\n');
%FINAL RESULT ENCODER
%FINAL_RESULT = (1:450);

[ENCODE,indx] = BIT_PROCESSING(bitsequence,Index-
1,finalSequence);
for i=1:1:450+256+256+256+256+256+256+256
    FINAL_RESULT(i)=char(ENCODE(i));
end

```

### **BIT PROCESSING**

```

function [RESULT,INDEX] = BIT_PROCESSING( Buffer, bytelength, finalSequence )
    BitSequence = zeros(size(bytelength*8+1));
    BitSequenceStuffed = zeros(size(bytelength*8+bytelength*8/5+1));
    for i=1:1:(bytelength*8+1)
        BitSequence(i) = 0;
    end

```

```

for i=1:1:(bytlength*8+bytlength*8/5+1)
    BitSequenceStuffed(i) = 0;
end

k = 1; %general counter
size_ = 1;
%s = 0; %stuffed sequence counter
cnt = 0;%Bit stuff counter
remBits = 0;
%temp = 0;
byte_temp=zeros(size(255*8));%max message lenght 255 bytes
for i=1:1:(255*8)
    byte_temp(i) = 0;
end
%Convert bits to byte size
%for (int i = 0; i < bytlength ; i++)
% {
%for (register uint8_t t=128; t>0 ; t = t/2) {
%if (Buffer[i] & t) BitSequence[k++] = 0x01;
%else BitSequence[k++] = 0x00;
% }
% }

%k =1;
arr_nums = [128,64,32,16,8,4,2,1];
for i=1:1:bytlength
    for x=1:1:8
        if bitand(Buffer(i),arr_nums(x))
            BitSequence(k) = 1;%hex2dec('01');
        else
            BitSequence(k) = 0;%hex2dec('00');
        end
        k=k+1;
    end
end
end

fprintf('Size:');
fprintf(num2str(k-1));
fprintf('\n');
%stuff a 0 after five consecutive 1s.
%for (int i = 0; i < k ; i++)
% {
%if (BitSequence[i] == 0x01) cnt++;
%else cnt = 0; // restart count at 1
%BitSequenceStuffed[s++] = BitSequence[i]; // add the bit to the final sequence
%if (cnt == 5) // there are five consecutive bits of the same value

```

```

% {
%BitSequenceStuffed[s++] = 0x00; // stuff with a zero bit
%cnt = 0; // and reset cnt to zero
% }
% }
s = 1;
c = 1;
for i = 1:1:k-1
    if BitSequence(i) == 1 %hex2dec('01')
        cnt = cnt+1;
    else
        cnt = 0;
    end
    BitSequenceStuffed(s) = BitSequence(i);
    %fprintf(num2str(BitSequenceStuffed(s)));
    %fprintf(' ');
    s = s+1;
    if cnt == 5
        BitSequenceStuffed(s) = 0;%hex2dec('00');
        %fprintf(num2str(BitSequenceStuffed(s)));
        %fprintf(' ');
        s=s+1;
        cnt = 0;
    end
end

end
for i=1:1:64
    Buffer(size_)= 0;%hex2dec('00');
    size_ = size_ + 1;
    for j = 1:1:6
        Buffer(size_)=1;%hex2dec('01');
        size_ = size_ + 1;
    end
    Buffer(size_)=0;%hex2dec('00');
    size_ = size_ + 1;
end
max= s-1;
%for (int i=0; i < s ; i++) Buffer[_size++] = BitSequenceStuffed[i];
for i = 1:1:max;
    Buffer(size_) = BitSequenceStuffed(i);
    size_ = size_+1;
end

%Insert 0b01111110 (FLAG)
%Buffer[_size++] = 0x00;
Buffer(size_) = 0;%hex2dec('00');

```

```

size_ = size_+1;
%for (int j=0; j < 6 ; j++)
%    {
%        Buffer[_size++] = 0x01;
%    }
%    Buffer[_size++] = 0x00;
%
for i=1:1:6
    Buffer(size_) = 1;%hex2dec('01');
    size_ = size_+1;
end
Buffer(size_) = 0;%hex2dec('00');
size_ = size_+1;

for i =1:1:((255*8))
    byte_temp(i) = 0;%hex2dec('00');
end

%for (int i = 0; i < 255*8 ; i++) byte_temp[i] = 0x00;
%NRZI encoding
%for (int i=0; i < _size ; i++)
%{
%if (Buffer[i] == 0x00)
%{
%byte_temp[i+1] = ! byte_temp[i];
%}
%else
%{
%byte_temp[i+1] = byte_temp[i];
%}
%}
for i=1:1:size_-1
    if Buffer(i)== 0
        byte_temp(i+1) = ~(byte_temp(i));
    else
        byte_temp(i+1) = (byte_temp(i));
    end
    fprintf(num2str(byte_temp(i)));
    fprintf(' ');
end
fprintf('\n');

%extrabits = (_size+1) % 8;
%if (((_size+1) % 8) > 0) remBits = 8 - ((_size+1) % 8);
if mod(size_,8) > 0
    remBits = 8 - (mod(size_,8));

```

```

end
fprintf('Rembits:');
fprintf(num2str(remBits));
fprintf('\n');
%for (int i = (_size + 1) ; i < (_size + 1 + remBits ) ; i++)
%{
%byte_temp[i] = 0x01;
%}

for i=size_+1:1:size_+ 1 + remBits
    byte_temp(i) = 1;% hex2dec('01');

end

%Convert to bit after NRZI and added remaining bits to form byte array
Index = 1;
%fprintf('BYTE TEMP:');
%fprintf(num2str(byte_temp));
fprintf('\n');
fprintf('Final sequence:\n');
for i = 1:8:(size_+ remBits)
    %fprintf(num2str(i));
    %fprintf(' ');
    temp = 0;
    if byte_temp(i) == 1 %hex2dec('01')
        temp = temp + bin2dec('10000000');
    end
    if byte_temp(i+1) == 1 %hex2dec('01')
        temp = temp + bin2dec('01000000');
    end
    if byte_temp(i+2) == hex2dec('01')
        temp = temp + bin2dec('00100000');
    end
    if byte_temp(i+3) == hex2dec('01')
        temp = temp + bin2dec('00010000');
    end
    if byte_temp(i+4) == hex2dec('01')
        temp = temp + bin2dec('00001000');
    end
    if byte_temp(i+5) == hex2dec('01')
        temp = temp + bin2dec('00000100');
    end
    if byte_temp(i+6) == hex2dec('01')
        temp = temp + bin2dec('00000010');
    end
    if byte_temp(i+7) == hex2dec('01')

```

```

    temp = temp + bin2dec('00000001');
end
finalSequence(Index) = temp;
%fprintf(num2str(Index));
%fprintf(' ');
fprintf(num2str(finalSequence(Index)));
fprintf(' ');
Index = Index+1;
end
fprintf('\n');
%fprintf(num2str(finalSequence));
%fprintf('\n');
fprintf('===== \n');
%fprintf('byte_temp : ');
%fprintf(num2str(byte_temp));
%fprintf('\n');
RESULT = finalSequence;
INDEX = Index-1;

```

### **MSB\_TO\_LSB\_8BIT**

```

function y = MSB_LSB_swap_8bit(v)
    m = '55'; %'0x55';
    n = '33';
    j = '0F';
    %swap odd and even bits
    %v = ((v >> 1) & 0x55) | ((v & 0x55) << 1);
    t = bitshift(v,-1);
    t = bitand(t,hex2dec(m));
    s = bitand(v, hex2dec(m));
    s = bitshift(s,1);
    v = bitor(t,s);
    %swap consecutive pairs
    %v = ((v >> 2) & 0x33) | ((v & 0x33) << 2);
    t = bitshift(v,-2);
    t = bitand(t,hex2dec(n));
    s = bitand(v, hex2dec(n));
    s = bitshift(s,2);
    v = bitor(t,s);
    %swap nibbles ...
    %v = ((v >> 4) & 0x0F) | ((v & 0x0F) << 4);
    t = bitshift(v,-4);
    t = bitand(t,hex2dec(j));
    s = bitand(v, hex2dec(j));
    s = bitshift(s,4);
    y = bitor(t,s);
end

```

### LAMPIRAN 3

#### Listing Program matlab FER, CER, dan BER

```
tr=('Jumlah Karakter Pengirim');
re=('Jumlah Karakter Penerima');
fe=[];
for i=1:(length(tr)/250)
    if tr(1+250*(i-1):250*i)==re(1+250*(i-1):250*i)
        fe=[fe 0];
    else
        fe=[fe 1];
    end
end
FER=sum(fe)/length(fe)
ce=[];
for i=1:length(tr)
    if re(i)==tr(i)
        ce=[ce 0];
    else
        ce=[ce 1];
    end
end
CER=sum(ce)/length(ce)
tr=dec2bin(tr);
re=dec2bin(re);
tr1=[num2str(zeros(9,1)) tr];
re1=[num2str(zeros(9,1)) re];
be=[];
for i=1:72
    if re1(i)==tr1(i)
        be=[be 0];
    else
        be=[be 1];clc
    end
end
ber=sum(be)/length(be)
```

## LAMPIRAN 4

### Listing Program Matlab Pengolahan Data BER, CER, FER dan SNR

```
%Plot Analisis BER CER FER dan SNR
close all

figure(1)
plot(DataBER(:,1),DataBER(:,2),'-ro');
hold on
plot(DataBER(:,1),DataBER(:,3),'-.*');
title('Grafik BER Tanpa Protocol AX.25 Vs Dengan Protocol
AX.25');
xlabel('Jumlah Karakter');ylabel('BER Rata-Rate')
legend('Tanpa Protocol AX.25','Dengan Protocol AX.25')

figure(2)
plot(DataCER(:,1),DataCER(:,2),'-ro');
hold on
plot(DataCER(:,1),DataCER(:,3),'-.*');
title('Grafik CER Tanpa Protocol AX.25 Vs Dengan Protocol
AX.25');
xlabel('Jumlah Karakter');ylabel('CER Rata-Rata')
legend('Tanpa Protocol AX.25','Dengan Protocol AX.25')

figure(3)
plot(DataFER(:,1),DataFER(:,2),'-ro');
hold on
plot(DataFER(:,1),DataFER(:,3),'-.*');
title('Grafik FER Tanpa Protocol AX.25 Vs Dengan Protocol
AX.25');
xlabel('Total Karakter');ylabel('FER Rata-Rata')
legend('Tanpa Protokol AX.25',' Dengan Protocol AX.25')

figure(4)
plot(DataSNR(:,1),DataSNR(:,2),'-ro');
hold on
plot(DataSNR(:,1),DataSNR(:,3),'-.*')
title('Grafik SNR Tanpa Protocol AX.25 Vs Dengan Protocol
AX.25');
xlabel('Jumlah Karakter');ylabel('SNR(dB) Rata-Rata')
legend('Tanpa Protocol AX.25',' Dengan Protocol AX.25')
```



## LAMPIRAN 5

### Data hasil Pengukuran

#### Tanpa Protocol :

##### 500 KARAKTER

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,51E-03	5,77E-05	130,00	21,14
2	0,5	0,02	0,4517	6,72E-03	8,04E-05	83,55	19,22
3	1	0,02	0,454	7,15E-03	7,86E-05	90,96	19,59
4	0,5	0,066	0,034	1,05E-01	1,17E-03	89,79	19,53
5	0	0	0	7,52E-03	7,06E-05	106,52	20,27
6	0	0	0	7,53E-03	5,80E-05	129,81	21,13
7	0	0	0	8,03E-03	6,59E-05	121,85	20,86
8	0	0	0	6,88E-03	7,94E-05	86,64	19,38
9	0,5	0,028	0,0147	7,61E-03	7,08E-05	107,42	20,31
10	0	0	0	7,51E-03	6,23E-05	120,55	20,81
<b>RATA-RATA</b>	<b>0,25</b>	<b>0,0134</b>	<b>0,09544</b>	<b>1,72E-02</b>	<b>1,80E-04</b>	<b>106,71</b>	<b>20,22</b>

##### 700 KARAKTER

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,52E-03	9,18E-05	81,94	19,13
2	0,5	0,02	0,005	7,75E-03	1,10E-04	70,22	18,46
3	0	0	0	9,06E-03	1,42E-04	63,95	18,06
4	0	0	0	7,38E-03	6,70E-06	1102,57	30,42
5	0	0	0	7,26E-03	1,03E-04	70,38	18,47
6	0	0,0186	0,0163	8,62E-03	1,63E-04	53,06	17,25
7	0,5	0,0171	0,0138	8,96E-03	1,26E-04	71,05	18,52
8	0	0,0186	0,0138	7,56E-03	2,25E-06	3355,85	35,26
9	0	0	0	7,49E-03	2,89E-06	2594,92	34,14
10	0	0	0	7,82E-03	1,23E-05	634,96	28,03
<b>RATA-RATA</b>	<b>0,1</b>	<b>0,00743</b>	<b>0,00489</b>	<b>7,94E-03</b>	<b>7,60E-05</b>	<b>809,89</b>	<b>23,77</b>

**1000 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,14E-03	1,32E-04	54,16	17,34
2	0,25	0,114	0,0461	7,41E-03	8,39E-05	88,35	19,46
3	0,75	0,076	0,0367	7,57E-03	7,01E-05	108,06	20,34
4	0,25	0,009	0,45	7,71E-03	5,95E-05	129,55	21,12
5	0,25	0,009	0,4484	7,84E-03	1,99E-05	393,88	25,95
6	0	0	0	7,82E-03	5,72E-05	136,71	21,36
7	0	0	0	7,68E-03	5,06E-06	1518,34	31,81
8	0,25	0,009	0,4494	6,66E-03	6,16E-05	108,06	20,34
9	0	0	0	7,66E-03	8,78E-05	87,31	19,41
10	0,75	0,084	0,041	7,66E-03	8,78E-05	87,31	19,41
<b>RATA-RATA</b>	<b>0,25</b>	<b>0,0301</b>	<b>0,14716</b>	<b>7,52E-03</b>	<b>6,65E-05</b>	<b>271,17</b>	<b>21,65</b>

**1500 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,45E-03	7,62E-05	97,73	19,90
2	0	0	0	7,64E-03	6,15E-05	124,12	20,94
3	0,1667	0,0067	0,4497	8,80E-03	1,04E-04	84,85	19,29
4	0,5	0,4853	0,4494	7,44E-03	3,75E-06	1983,79	32,97
5	1	0,2373	0,0927	7,49E-03	5,47E-06	1368,51	31,36
6	0,667	0,102	0,4498	7,43E-03	8,71E-05	85,39	19,31
7	0,3333	0,0087	0,4505	7,43E-03	8,71E-05	85,39	19,31
8	0	0	0	7,39E-03	1,96E-04	37,66	15,76
9	0	0	0	7,36E-03	8,54E-05	86,20	19,35
10	1	0,232	0,0927	7,14E-03	1,32E-04	54,16	17,34
<b>RATA-RATA</b>	<b>0,3667</b>	<b>0,1072</b>	<b>0,19848</b>	<b>7,56E-03</b>	<b>8,38E-05</b>	<b>400,78</b>	<b>21,55</b>

**2000 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,81E-03	1,37E-05	570,9713	27,57
2	0,375	0,025	0,0127	8,14E-03	3,54E-05	230,27292	23,62
3	0,125	0,0045	0,4486	7,38E-03	2,35E-05	313,63188	24,96
4	0,125	0,0075	0,0039	6,78E-03	7,04E-05	96,293018	19,84
5	0,25	0,011	0,4483	7,38E-03	1,07E-04	69,128244	18,40
6	0,375	0,013	0,4482	7,50E-03	5,95E-05	126,11766	21,01
7	0,375	0,036	0,4476	6,68E-03	6,14E-05	108,87299	20,37
8	0	0	0	7,49E-03	5,57E-05	134,45921	21,29
9	0,25	0,0245	0,0125	7,39E-03	8,30E-05	88,972562	19,49
10	0,25	0,0115	0,448	7,49E-03	6,36E-05	117,7402	20,71
<b>RATA-RATA</b>	<b>0,2125</b>	<b>0,0133</b>	<b>0,22698</b>	<b>7,40E-03</b>	<b>5,73E-05</b>	<b>185,65</b>	<b>21,72</b>

Dengan Protocol:

**500 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,81E-03	1,74E-05	448,34	26,52
2	0	0	0	8,85E-03	6,00E-05	147,63	21,69
3	0	0	0	7,34E-03	4,19E-06	1751,54	32,43
4	0	0	0	7,81E-03	6,79E-06	1149,88	30,61
5	0	0	0	7,83E-03	1,37E-05	572,30	27,58
6	0	0	0	7,85E-03	6,56E-06	1196,04	30,78
7	0	0	0	7,72E-03	8,36E-06	923,75	29,66
8	0	0	0	8,61E-03	8,60E-05	100,04	20,00
9	0	0	0	8,10E-03	3,36E-05	240,95	23,82
10	0	0	0	7,57E-03	3,61E-05	209,48	23,21
<b>RATA-RATA</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>7,95E-03</b>	<b>2,73E-05</b>	<b>674,00</b>	<b>26,63</b>

**700 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,69E-03	1,59E-05	483,79	26,85
2	0	0	0	7,92E-03	2,58E-05	307,31	24,88
3	0	0	0	7,37E-03	2,47E-06	2988,72	34,75
4	0	0	0	7,96E-03	1,43E-05	554,59	27,44
5	0	0	0	8,78E-03	8,55E-05	102,63	20,11
6	0	0	0	7,62E-03	2,11E-06	3616,58	35,58
7	0	0	0	7,09E-03	3,91E-06	1812,20	32,58
8	0	0	0	7,97E-03	5,95E-05	133,94	21,27
9	0	0	0	8,05E-03	1,67E-05	481,69	26,83
10	0	0	0	8,04E-03	1,83E-05	439,58	26,43
<b>RATA-RATA</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>7,85E-03</b>	<b>2,45E-05</b>	<b>1092,10</b>	<b>27,67</b>

**1000 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,35E-03	3,94E-06	1865,15	32,71
2	0	0	0	7,21E-03	9,82E-05	73,40	18,66
3	0	0	0	8,95E-03	1,47E-04	61,06	17,86
4	0	0	0	8,43E-03	2,90E-05	290,66	24,63
5	0	0	0	8,11E-03	1,38E-05	588,91	27,70
6	0	0	0	7,28E-03	3,93E-06	1852,86	32,68
7	0	0	0	7,43E-03	1,91E-05	389,14	25,90
8	0	0	0	7,30E-03	1,01E-06	7216,66	38,58
9	0	0	0	7,12E-03	1,23E-04	57,79	17,62
10	0	0	0	7,54E-03	2,10E-05	358,48	25,54
<b>RATA-RATA</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>7,67E-03</b>	<b>4,60E-05</b>	<b>1275,41</b>	<b>26,19</b>

**1500 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,45E-03	1,88E-06	3960,38	35,98
2	0	0	0	7,44E-03	2,16E-05	344,47	25,37
3	0	0	0	7,53E-03	1,47E-05	513,05	27,10
4	0	0	0	7,60E-03	5,90E-05	128,84	21,10
5	0	0	0	7,63E-03	6,52E-05	117,00	20,68
6	0	0	0	7,64E-03	1,49E-05	513,17	27,10
7	0	0	0	8,06E-03	6,55E-05	123,06	20,90
8	0	0	0	8,44E-03	4,66E-05	180,95	22,58
9	0	0	0	7,90E-03	1,50E-05	525,35	27,20
10	0	0	0	6,66E-03	2,61E-06	2551,85	34,07
<b>RATA-RATA</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>7,64E-03</b>	<b>3,07E-05</b>	<b>895,81</b>	<b>26,21</b>

**2000 KARAKTER**

PENGUJIAN KE-N	FER	CER	BER	SIGNAL POWER	NOISE POWER	SNR	SNR (dB)
1	0	0	0	7,15E-03	1,40E-04	51,02	17,08
2	0	0	0	7,43E-03	1,21E-05	613,89	27,88
3	0	0	0	7,26E-03	6,15E-07	11794,24	40,72
4	0	0	0	7,15E-03	1,09E-04	65,28	18,15
5	0	0	0	7,78E-03	7,50E-06	1036,48	30,16
6	0	0	0	7,39E-03	5,76E-06	1282,10	31,08
7	0	0	0	7,64E-03	1,06E-04	72,34	18,59
8	0	0	0	7,50E-03	6,76E-07	11088,72	40,45
9	0	0	0	7,35E-03	2,89E-06	2542,98	34,05
10	0	0	0	7,56E-03	6,58E-05	114,90	20,60
<b>RATA-RATA</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>7,42E-03</b>	<b>4,51E-05</b>	<b>2866,19</b>	<b>27,88</b>

**Kesimpulan :**

**FER Rata-Rata**

KARAKTER	FER RATA-RATA	
	TP	DP
500	0	0
700	0,25	0
1000	1	0
1500	0,25	0
2000	0,2125	0

**CER Rata-Rata**

KARAKTER	CER RATA-RATA	
	TP	DP
500	0,0134	0
700	0,00743	0
1000	0,0301	0
1500	0,1072	0
2000	0,0133	0

**BER Rata-Rata**

KARAKTER	BER RATA-RATA	
	TP	DP
500	0,09544	0
700	0,00489	0
1000	0,14716	0
1500	0,19848	0
2000	0,22698	0

**SNR (dB) Rata-Rata**

KARAKTER	SNR (dB) RATA-RATA	
	TP	DP
500	20,22	26,63
700	23,77	27,67
1000	21,65	26,19
1500	21,55	26,21
2000	21,72	27,88

## DAFTAR RIWAYAT

Penulis telah menyelesaikan pendidikan jenjang S1 di Sekolah Tinggi Teknologi Telematika Telkom Purwokerto dengan jurusan Teknik Telekomunikasi, lulus pada Februari 2017 dan terdaftar sebagai mahasiswa Departemen Teknik Elektro Program Pascasarjana Strata Dua (S2) tahun 2018 pada bidang keahlian Telekomunikasi Multimedia di Institut Teknologi Sepuluh Nopember (ITS) Surabaya.



Penulis telah mengikuti seminar proposal tesis dengan judul **“PENGEMBANGAN DAN EVALUASI DIGITAL HF BERBASIS SOFTWARE-DEFINED RADIO”** pada 29 Mei 2019 dan ujian tesis pada Januari 2020 sebagai salah satu syarat untuk memperoleh gelar Magister Teknik (M.T).

Nama : Agung Wicaksono

Alamat : Perum Griya Tegal Sari Indah Jl. Pandawa G2-3 RT 004  
RW 009 Dukuhwaluh, Kembaran, Banyumas, Jawa Tengah.

Tempat/Tanggal lahir : Banyumas, 14 Mei 1995

Email : agungwicaksono530@gmail.com

Riwayat Pendidikan :

1. SDN 1 Dukuhwaluh (2001-2007)
2. SMP Negeri 7 Purwokerto (2007-2010)
3. SMA Negeri 4 Purwokerto (2010-2013)
4. Sekolah Tinggi Teknologi Telematika Telkom (2013-2017)
5. Institut Teknologi Sepuluh Nopember (2018-2020)

Selama menjadi mahasiswa Departemen Teknik Elektro bidang keahlian Telekomunikasi Multimedia Program Pascasarjana Strata Dua (S2) Institut Teknologi Sepuluh Nopember (ITS) Surabaya, penulis mendapatkan prestasi sebagai **“Honorary Mention”** di 2019 IEEE Communication Society Student Competition dengan mendapatkan peringkat 15 dari 51 peserta Dunia.