



TUGAS AKHIR - TE 141599

**IMPLEMENTASI DAN ANALISA KINERJA SISTEM
KOMUNIKASI KOOPERATIF *DECODING AND FORWARD*
(DF) DENGAN SKEMA *NETWORK CODING*
MENGUNAKAN *WIRELESS OPEN-ACCESS RESEARCH
PLATFORM (WARP)***

**Diefa Agung Riyadi
NRP. 2213106007**

**Dosen Pembimbing
Dr. Ir. Suwadi, M.T.
Dr. Ir. Wirawan, DEA.**

**JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016**



FINAL PROJECT - TE 141599

**IMPLEMENTATION AND PERFORMANCE ANALYSIS OF
DECODING AND FORWARD (DF) COOPERATIVE
COMMUNICATION SYSTEM WITH NETWORK CODING
SCHEME USING WIRELESS OPEN-ACCESS RESEARCH
PLATFORM (WARP)**

**Diefa Agung Riyadi
NRP. 2213106007**

Supervisors

Dr. Ir. Suwadi, M.T.

Dr. Ir. Wirawan, DEA.

**DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

**IMPLEMENTASI DAN ANALISA KINERJA
SISTEM KOMUNIKASI KOOPERATIF
DECODING AND FORWARD (DF)
DENGAN SKEMA NETWORK CODING MENGGUNAKAN
WIRELESS OPEN-ACCESS RESEARCH PLATFORM (WARP)**


TUGAS AKHIR

**Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Telekomunikasi Multimedia
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

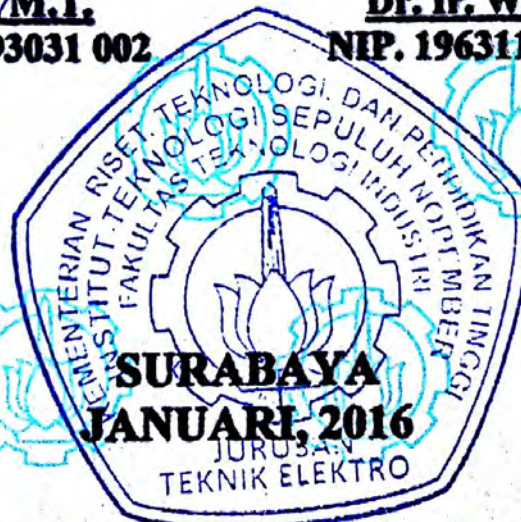
Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II


Dr. Ir. Suwadi, M.T.
NIP. 19680818 1993031 002


Dr. Ir. Wirawan, DEA.
NIP. 19631109 1989031 011



Implementasi dan Analisa Kinerja Sistem Komunikasi Kooperatif *Decoding and Forward (DF)* dengan Skema *Network Coding* Menggunakan *Wireless Open-Access Research Platform (WARP)*

Nama : Diefa Agung Riyadi
Dosen Pembimbing : 1. Dr. Ir. Suwadi, M.T.
2. Dr. Ir. Wirawan, DEA.

ABSTRAK

Network Coding adalah teknik dimana data yang dikirimkan akan dikodekan di *relay* dalam bentuk XOR dan diterjemahkan kembali. Salah satu tujuan dari *Network Coding* adalah peningkatan *throughput*. Sedangkan sistem komunikasi kooperatif merupakan komunikasi yang memanfaatkan penggunaan untuk mengeliminasi *fading* untuk meningkatkan kinerja sistem, skema *decoding and forward* pada sistem komunikasi kooperatif adalah informasi akan dikodekan terlebih dahulu sebelum dikirimkan ke tujuan. Penggunaan *network coding* pada sistem komunikasi kooperatif dimaksudkan untuk meningkatkan kinerja dari sistem komunikasi tersebut. *Wireless Open-Access Research Platform (WARP)* adalah salah satu *software defined radio* yang dikembangkan untuk mengimplementasikan sistem komunikasi. Pada tugas akhir ini, sistem komunikasi kooperatif yang akan digabungkan dengan teknik *network coding* akan disimulasikan pada Matlab dan diimplementasikan pada WARP. Dari hasil implementasi yang didapat menunjukkan bahwa komunikasi *network coding* mempunyai kinerja yang tidak jauh berbeda bila diukur berdasarkan BER seperti pada saat diukur dengan daya pancar -20.3 dBm nilai BER untuk komunikasi kooperatif 0 dan kooperatif *network coding* 0.0002, namun jika diukur berdasarkan *throughput* kinerja komunikasi kooperatif *network coding* lebih baik dibandingkan kooperatif konvensional, seperti pada saat daya pancar berada pada level -17.1 dBm *throughput* yang dihasilkan kooperatif *network coding* adalah 2.6 kbps, sementara untuk kooperatif konvensional adalah 2.2 kbps.

Kata Kunci : Network Coding, Decoding and Forward, Software Defined Radio, WARP

Implementation and Performance Analisis of Decoding and Forward (DF) Cooperative Communication System with Network Coding Scheme Using Wireless Open-Access Research Platform (WARP)

Name : Diefa Agung Riyadi
Supervisors : 1. Dr. Ir. Suwadi, M.T.
2. Dr. Ir. Wirawan, DEA.

ABSTRACK

Network Coding is a technique where the data transmitted will be encoded on relay first in the form XOR and will be translated again. One of purpose of network coding is improve throughput. Whereas Cooperative Communication System is Communication that use relay for eliminate fading to improve system performance, decoding and forward scheme on Cooperative Communication System is scheme that will be encode message first before sent it to destination. The purpose of using network coding on cooperative communication system is for improving performance the system. Wireless Open-Access Research Platform (WARP) is one of software defined radio which was developed for implementing the communication system. In this final project cooperative communication which will be combined with network coding technique will be simulated on matlab and implemented on WARP. From the implementation result, performance of cooperative communication system with network coding will be analyzed. From the implementation result is showing that cooperative communication with network coding isn't much different with conventional cooperative communication with BER parameters, such as when measuring with power transmit -20.3 dBm, the values of bit error rate for cooperative with power transmit is 0 and cooperative network coding is 0.0002 , but if measuring based on throughput, performance of cooperative communication with network coding is better than conventional cooperative communication, such as when power transmit at -17.1 dBm, throughput for cooperative network coding is 2.6 kbps, and for conventional cooperative is 2.2 kbps .

Keywords : Network Coding, Decoding and Forward, Software Defined Radio, WARP

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakaatuh

Alhamdulillah, segala puji dan syukur penulis panjatkan kehadirat Allah SWT, karena atas nikmat dan karunianya penulis dapat menyelesaikan tugas akhir yang berjudul :

Implementasi dan Analisa Kinerja Sistem Komunikasi Kooperatif *Decoding and Forward* (DF) dengan Skema *Network Coding* Menggunakan *Wireless Open-Access Research Platform* (WARP).

Tugas akhir ini dibuat sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik pada jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya. Selama penelitian penulis mendapatkan banyak dorongan, saran, bimbingan dan arahan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih kepada :

1. Ayahanda, Darsono dan Ibunda, Sarinah yang ada di Jakarta, yang selalu mendoakan, memberi saran, menghibur dan memberi bantuan secara materil dan non-materil kepada penulis.
2. Bapak Dr. Ir. Suwadi, MT, dan Bapak Dr. Ir. Wirawan, DEA. yang memberikan arahan dan bimbingan secara teknis selama proses penelitian
3. Keluarga TMM LJ 2013 Genap (Sherly, Prastiyanto, Dinar, Bambang, Ummul, Banyu, Prasetyo, Dwi, Aniko, Dessy, Ranieta, Tania, Dita) terimakasih atas canda tawanya selama kuliah walaupun kadang nggak berkualitas
4. Warga Lab Komunikasi Multimedia B304 khususnya untuk "*Network Coding Research Team*"(Pak Iqbal, Mbak Anin, Tiyen, Bambang, Ummul) terimakasih atas bantuan dan sharing ilmunya
5. Kakakku Bayu Ardiansyah terimakasih atas bantuannya selama penulis menjalani kuliah di ITS, dan Wira Indani, ST., MT. terimakasih atas transfer ilmunya untuk menyelesaikan tugas akhir ini

Penulis menyadari bahwa dalam karya tulis ini masih banyak kekurangan sehingga diperlukan kritik dan saran untuk mengembangkan dan menyempurnakannya. Akhir kata penulis mengucapkan terima kasih.

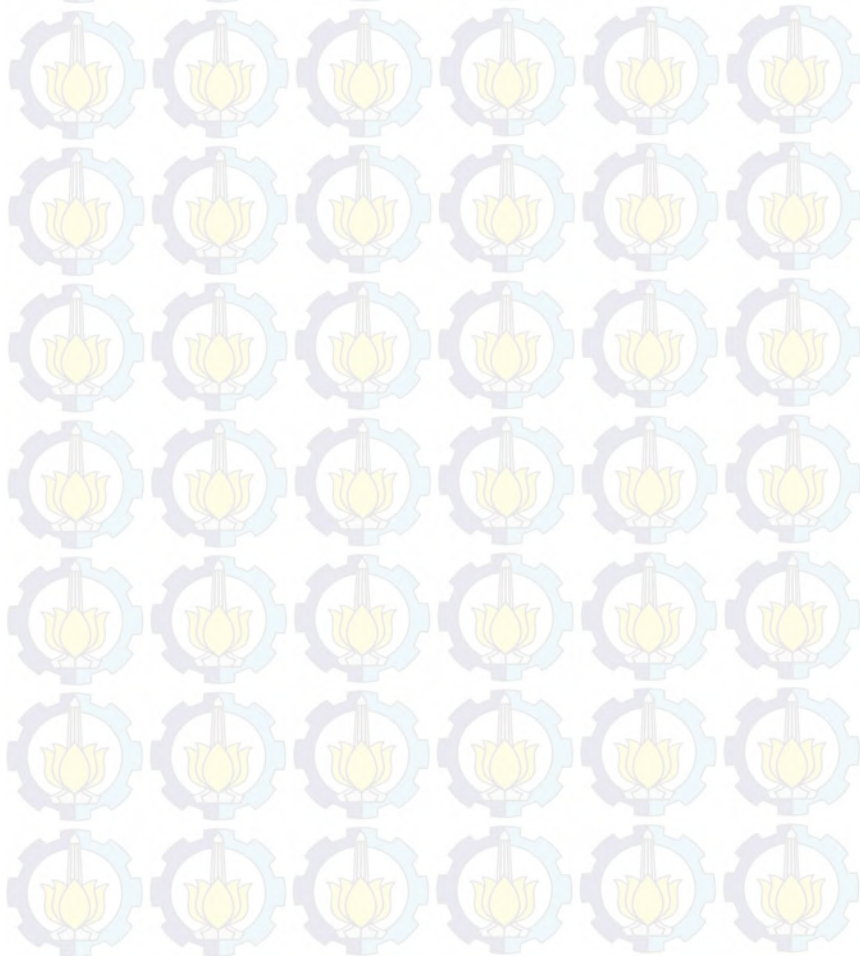
Surabaya, Januari 2016
Diefa Agung Riyadi

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	5
1.7 Manfaat Tugas Akhir	5
BAB 2 TINJAUAN PUSTAKA	7
2.1 Sistem Komunikasi Nirkabel	7
2.2 Network Coding	7
2.2.1 Skema Tradisional Tanpa <i>Network Coding</i>	8
2.2.2 Skema Straight Network Coding	8
2.2.3 Skema Physical Network Coding	9
2.3 Sistem Komunikasi Kooperatif	10
2.4 Teknik Forward	11
2.4.1 Amplify and Forward	11
2.4.2 Decode and Forward	11
2.5 Teknik Combining	13
2.5.1 Selective Combining	13
2.5.2 Equal Gain Combining	14
2.5.3 Maximun Ratio Combining	14
2.6 Kode Konvolusi	14
2.6.1 Diagram <i>State</i>	17
2.6.2 Diagram <i>Tree</i>	18
2.6.3 Diagram Trellis	19
2.6.4 Algoritma Viterbi	19
2.7 Modulasi Digital	20
2.8 Software Defined Radio (SDR)	20
2.9 Wireless Open-Access Research Platform (WARP)	21
2.9.1 WARP Board	22

2.9.2	WARPLab	24
2.10	Receive Strength Signal Indicator (RSSI)	26
BAB 3 PERANCANGAN DAN IMPLEMENTASI		31
3.1	Alur Perancangan dan Implementasi Sistem	31
3.2	Simulasi Sistem Komunikasi	32
3.2.1	Skema Simulasi Sistem.....	32
3.2.2	Diagram Blok Simulasi Sistem.....	32
3.2.3	Hasil Simulasi Sistem Komunikasi.....	37
3.3	Blok Diagram Perancangan Sistem	40
3.3.1	Parameter Sistem Komunikasi	41
3.3.2	Proses pada Pengirim.....	41
3.3.3	Proses Transmisi pada WARP	42
3.3.4	Proses pada Penerima	43
3.4	Integrasi Laptop/PC ke Perangkat WARP.....	44
3.5	Skenario Pengukuran.....	45
3.5.1	Pengukuran <i>Indoor</i>	46
3.5.2	Pengukuran <i>Outdoor</i>	47
3.5.3	Kondisi Pengukuran.....	48
3.6	Perhitungan <i>Bit Error Rate</i> (BER)	48
3.7	Perhitungan <i>Throughput</i>	49
BAB 4 HASIL DAN ANALISA		51
4.1	Skema Implementasi	51
4.2	Analisa Kinerja <i>Network Coding</i> pada Sistem Komunikasi Kooperatif (<i>Indoor</i>).....	52
4.2.1	Line of Sight	52
4.2.2	Non Line of Sight	55
4.3	Analisa Kinerja <i>Network Coding</i> pada Sistem Komunikasi Kooperatif (<i>Outdoor</i>).....	57
4.3.1	Line of Sight	57
4.3.2	Non Line of Sight	61
4.4	Analisa Kinerja <i>Network Coding</i> pada Sistem Komunikasi Kooperatif Berdasarkan Jarak.....	64
4.4.1	Perbandingan Kinerja pada Kondisi <i>Indoor</i>	64
4.4.2	Perbandingan Kinerja pada Kondisi <i>Outdoor</i>	67
4.5	Perbandingan <i>Throughput</i> Antara Sistem Komunikasi Kooperatif Dengan dan Tanpa <i>Network Coding</i>	70
BAB 5 PENUTUP		77
5.1	Kesimpulan.....	77
5.2	Saran.....	78

DAFTAR PUSTAKA	79
LAMPIRAN A : PENGESAHAN TUGAS AKHIR	81
LAMPIRAN B : DOKUMENTASI PENGUKURAN	83
LAMPIRAN C : LISTING PROGRAM	85
LAMPIRAN D : SETUP WARP	115
BIOGRAFI PENULIS	119





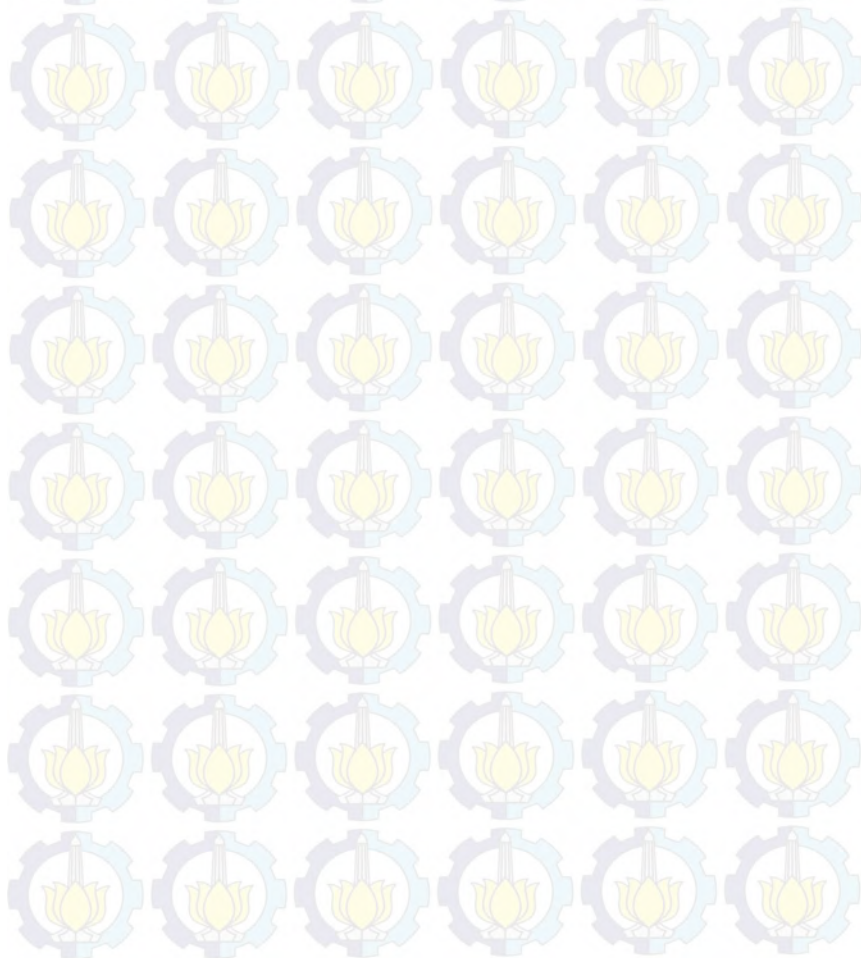
[Halaman Ini Sengaja Dikosongkan]

CONTENTS

ABSTRAK	i
ABSTRACT	iii
PREFACE	v
CONTENTS	vii
FIGURE	xi
TABLE	xiii
CHAPTER 1 PERLIMINARY	1
1.1 Background	1
1.2 Problem Formula	2
1.3 Problem Constraint	2
1.4 Purposes	2
1.5 Methodology	2
1.6 Writing Systematics	5
1.7 Benefit	5
CHAPTER 2 THEORY	7
2.1 Wireless Communication System	7
2.2 Network Coding	7
2.2.1 Traditional Scheme without Network Coding	8
2.2.2 Straight Network Coding Scheme	8
2.2.3 Physical Network Coding Scheme	9
2.3 Cooperative Communication System	10
2.4 Forward Techniques	11
2.4.1 Amplify and Forward	11
2.4.2 Decode and Forward	11
2.5 Combining Techniques	13
2.5.1 Selective Combining	13
2.5.2 Equal Gain Combining	14
2.5.3 Maximun Ratio Combining	14
2.6 Convolutional Code	14
2.6.1 State Diagram	17
2.6.2 Tree Diagram	18
2.6.3 Trellis Diagram	19
2.6.4 Viterbi Algorithm	19
2.7 Digital Modulation	20
2.8 Software Defined Radio (SDR)	20
2.9 Wireless Open-Access Research Platform (WARP)	21
2.9.1 WARP Board	22

2.9.2	WARPLab	24
2.10	Receive Strength Signal Indicator (RSSI)	26
CHAPTER 3 DESIGN AND IMPLEMENTATION		31
3.1	Design and System Implementation Flow	31
3.2	Communication System Simulation	32
3.2.1	Simulation System Scheme	32
3.2.2	Block Diagram Simulation System.....	32
3.2.3	Communication System Simulation Result	37
3.3	Block Diagram Implementation System.....	40
3.3.1	Communication System Parameters	41
3.3.2	Transmission Process.....	41
3.3.3	Transmission Process on WARP	42
3.3.4	Receiver Process	43
3.4	Software Integration to WARP Device	44
3.5	Measuring Scenario	45
3.5.1	Indoor Measuring.....	46
3.5.2	Outdoor Measuring.....	47
3.5.3	Measuring Condition	48
3.6	Bit Error Rate Calculation	48
3.7	Throughput Calculation.....	49
CHAPTER 4 RESULT AND ANALYSIS		51
4.1	Implementation Scheme	51
4.2	Performance Analysis of Network Coding on Cooperative Communication System (Indoor).....	52
4.2.1	Line of Sight	52
4.2.2	Non Line of Sight	55
4.3	Performance Analysis of Network Coding on Cooperative Communication System (Outdoor)	57
4.3.1	Line of Sight	57
4.3.2	Non Line of Sight	61
4.4	Performance Analysis of Network Coding on Cooperative Communication System Based on Distance.....	64
4.4.1	Performance Comparison on Indoor Condition	64
4.4.2	Performance Comparison on Outdoor Condition	67
4.5	Throughput Comparison Between Cooperative Communication System with and without Network Coding	70
CHAPTER 5 CLOSING		77
5.1	Conclusion.....	77
5.2	Suggestion	78

BIBLIOGRAPHY	79
APPENDIX A : FINAL PROJECT APPROVAL	81
APPENDIX B : MEASURING DOCUMENTATION	83
APPENDIX C : PROGRAM LISTING	85
APPENDIX D : WARP SETUP	115
BIOGRAPHY	119





[Halaman Ini Sengaja Dikosongkan]

DAFTAR GAMBAR

Gambar 1.1 Alur Metodologi Penelitian.....	3
Gambar 2.1 <i>Butterfly Network</i>	7
Gambar 2.2 Skema Tradisional Tanpa <i>Network Coding</i>	8
Gambar 2.3 Skema <i>Straight Network Coding</i>	9
Gambar 2.4 <i>Physical Network Coding</i>	10
Gambar 2.5 Skema Sistem Komunikasi Kooperatif.....	10
Gambar 2.6 Skema Komunikasi Kooperatif <i>Amplify and Forward</i>	11
Gambar 2.7 Skema Komunikasi Kooperatif <i>Decode and Forward</i>	12
Gambar 2.8 Encoder Kode Konvolusi dengan <i>Coderate</i> $\frac{1}{2}$	17
Gambar 2.9 <i>Diagram State Rate</i> = $\frac{1}{2}$ dan $K = 3$	17
Gambar 2.10 <i>Diagram Tree Rate</i> = $\frac{1}{2}$ dan $K = 3$	18
Gambar 2.11 <i>Diagram Trellis Rate</i> = $\frac{1}{2}$ dan $K = 3$	19
Gambar 2.12 Arsitektur <i>Software Define Radio</i>	21
Gambar 2.13 Modul WARP Versi 2.2.....	22
Gambar 2.14 <i>Radio Board</i>	23
Gambar 2.15 Arsitektur <i>Radio Board WARP</i>	23
Gambar 2.16 Arsitektur WARPLab.....	24
Gambar 2.17 Penggunaan Kerangka Kerja WARPLab.....	26
Gambar 2.18 TX Output vs Gain Settings pada MAX2829.....	27
Gambar 2.19 Gain Rx Baseband pada MAX2829.....	28
Gambar 2.20 Grafik RSSI vs Daya Input pada MAX2829.....	29
Gambar 3.1 Diagram Alur Perancangan dan Implementasi Sistem.....	31
Gambar 3.2 Skema Simulasi Skema <i>Network Coding</i> pada Sistem Komunikasi Kooperatif DF.....	32
Gambar 3.3 Diagram Blok Simulasi Skema <i>Network Coding</i> pada Sistem Komunikasi Kooperatif DF.....	33
Gambar 3.4 Proses Pengiriman Data pada <i>Node Source</i>	34
Gambar 3.5 (a) Bit yang Dikirim dari <i>Source 1</i> , (b) Bit yang Dikirim dari <i>Source 2</i>	35
Gambar 3.6 Proses <i>Network Coding</i> pada <i>Node Relay</i>	35
Gambar 3.7 Data yang Diterima pada Relay Setelah Proses XOR.....	36
Gambar 3.8 Proses <i>Network Coding</i> pada <i>Node Destination</i>	36
Gambar 3.9 (a) Data yang Diterima <i>Node Destination 2</i> , (b) Data yang Diterima <i>Node Destination 1</i>	37
Gambar 3.10 Hasil Simulasi Kinerja <i>Bit Error Rate</i> Sistem Komunikasi Kooperatif DF dengan <i>Network Coding</i>	38

Gambar 3.11 Hasil Simulasi Kinerja <i>Throughput Network Coding</i> pada Sistem Komunikasi Kooperatif DF	39
Gambar 3.12 Diagram Blok Perancangan Skema <i>Network Coding</i> pada Sistem Komunikasi Kooperatif DF	40
Gambar 3.13 Integrasi PC/Laptop dengan Modul WARP	45
Gambar 3.14 Layout Ruangan AJ403	46
Gambar 3.15 <i>Layout</i> Pengukuran di Parkiran Dosen	47
Gambar 3.16 Sketsa Pengukuran <i>Line of Sight</i>	48
Gambar 3.17 Sketsa Pengukuran dengan Penghalang	48
Gambar 4.1 Skema Implementasi <i>Network Coding</i> pada Sistem Komunikasi Kooperatif DF	51
Gambar 4.2 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter	53
Gambar 4.3 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter	54
Gambar 4.4 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter	54
Gambar 4.5 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter	56
Gambar 4.6 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter	56
Gambar 4.7 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter	57
Gambar 4.8 Grafik Kurva Hasil Implementasi BER Jarak 30 Meter	59
Gambar 4.9 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter	59
Gambar 4.10 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter ..	60
Gambar 4.11 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter ..	60
Gambar 4.12 Grafik Kurva Hasil Implementasi BER Jarak 30 Meter	62
Gambar 4.13 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter ..	63
Gambar 4.14 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter ..	63
Gambar 4.15 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter ..	64
Gambar 4.16 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi LOS (<i>Indoor</i>)	65
Gambar 4.17 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi NLOS (<i>Indoor</i>)	67
Gambar 4.18 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi LOS (<i>Outdoor</i>)	68
Gambar 4.19 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi NLOS (<i>Outdoor</i>).....	70
Gambar 4.20 Grafik <i>Cyclic Time</i> pada Kondisi <i>Indoor</i>	71
Gambar 4.21 Grafik Kurva <i>Throughput</i> pada Kondisi <i>Indoor</i>	72
Gambar 4.22 Grafik <i>Cyclic Time</i> pada Kondisi <i>Outdoor</i>	74
Gambar 4.23 Grafik Kurva <i>Throughput</i> pada Kondisi <i>Outdoor</i>	75

DAFTAR TABEL

Tabel 3.1 Parameter Sistem Komunikasi.....	41
Tabel 3.2 Variasi Jarak Pengukuran <i>Indoor</i> (AJ403)	46
Tabel 3.3 Variasi Jarak Pengukuran <i>Outdoor</i> (Lapangan Parkir Dosen)	47
Tabel 4.1 Tabel Perbandingan BER Pada Jarak 8 Meter (<i>Indoor</i>)	52
Tabel 4.2 Tabel Perbandingan BER Pada Jarak 8 Meter (<i>Indoor</i>)	55
Tabel 4.3 Tabel Perbandingan BER Pada Jarak 30 Meter (<i>Outdoor</i>).....	58
Tabel 4.4 Tabel Perbandingan BER Pada Jarak 30 Meter (<i>Outdoor</i>)....	61
Tabel 4.5 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif <i>Network Coding</i> dengan Variasi Jarak pada Kondisi LOS (<i>Indoor</i>)	65
Tabel 4.6 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif <i>Network Coding</i> dengan Variasi Jarak pada Kondisi NLOS (<i>Indoor</i>) ...	66
Tabel 4.7 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif <i>Network Coding</i> dengan Variasi Jarak pada Kondisi LOS (<i>Outdoor</i>)....	68
Tabel 4.8 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif <i>Network Coding</i> dengan Variasi Jarak pada Kondisi NLOS (<i>Outdoor</i>)..	69
Tabel 4.9 Tabel Perbandingan <i>Cyclic Time</i> pada Sistem Komunikasi Kooperatif yang Memakai <i>Network Coding</i> dengan yang Tidak	71
Tabel 4.10 Tabel Perbandingan <i>Throughput</i> Antara Sistem Komunikasi yang Memakai <i>Network Coding</i> dengan yang Tidak (<i>Indoor</i>).....	72
Tabel 4.11 Tabel Perbandingan <i>Cyclic Time</i> pada Sistem Komunikasi Kooperatif yang Memakai <i>Network Coding</i> dengan yang Tidak	73
Tabel 4.12 Tabel Perbandingan <i>Throughput</i> Antara Sistem Komunikasi yang Memakai <i>Network Coding</i> dengan yang Tidak (<i>Outdoor</i>)	74



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Komunikasi nirkabel telah berkembang dalam beberapa tahun ini dikarenakan kemampuannya mendukung konektivitas dan akses *mobile*. Namun seiring dengan pergantian abad banyak upaya yang dilakukan untuk menciptakan komunikasi nirkabel dengan kehandalan dan *data rate* yang tinggi [1]. Sistem komunikasi kooperatif merupakan salah satu metode yang diterapkan dalam komunikasi nirkabel yang memungkinkan satu antenna didalam cakupan banyak pengguna untuk membagi antenna pengguna lain dengan memanfaatkan perangkatnya sehingga menghasilkan antenna *virtual* untuk mencapai *transmit diversity*. Dengan komunikasi kooperatif pengguna mungkin dapat meningkatkan efektifitas *quality of services* dengan *bit error rate* (BER) [2]. *Network coding* juga merupakan salah satu metode dalam sistem komunikasi nirkabel yang baru baru ini mendapat perhatian karena keuntungannya dalam meningkatkan *throughput* dan meningkatkan ketahanan untuk sistem multi-sumber. Prinsip dari *network coding* sendiri adalah menggabungkan beberapa informasi secara linier dan mentransmisikan ke dalam satu aliran, dengan dasar XOR skema *network coding* digunakan untuk melakukan pertukaran informasi antara sepasang *nodes* lewat dua *hops* [3]. Teknologi SDR diimplementasikan di beberapa fungsi pada system radio, seperti modulasi/demodulasi, pengolahan sinyal, pemrograman dan link-layer protocol pada software [4]. Salah satu jenis dari teknologi SDR yaitu *Wireless Open-Access Research Platform* (WARP). Platform WARP dikategorikan sebagai *platform* yang dapat mengimplementasikan algoritma *wireless* yang sangat kompleks [5]. WARP tersebut akan digunakan untuk mengimplementasikan *network coding* yang akan digabungkan dengan sistem komunikasi kooperatif dengan relay *decode and forward*, kemudian setelah itu akan dilakukan pengukuran untuk mendapatkan data data dan dianalisa untuk mengathui kelebihan dari *network coding* jika dikombinasikan dengan sistem komunikasi kooperatif

1.2 Perumusan Masalah

Masalah yang diharapkan untuk ditemukan solusinya melalui tugas akhir ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan *network coding* pada sistem komunikasi kooperatif ?
2. Bagaimana prinsip kerja dari metode *decoding and forward* pada implementasi sistem ?
3. Seberapa baik kinerja sistem komunikasi kooperatif jika ditambahkan *network coding* ?

1.3 Batasan Masalah

Untuk menyelesaikan permasalahan yang ada, maka dalam prakteknya akan ada pembatasan masalah seperti :

1. Metode yang dipakai pada sistem komunikasi kooperatif adalah *decoding and forward*
2. Parameter yang diukur adalah BER (*Bit Error Rate*) dan *Throughput*
3. Teknik *Combining* yang dipakai adalah *Selective Combining*
4. Modulasi yang digunakan adalah QPSK
5. WAPLab yang dipakai adalah versi 7
6. Matlab yang dipakai adalah R2013b

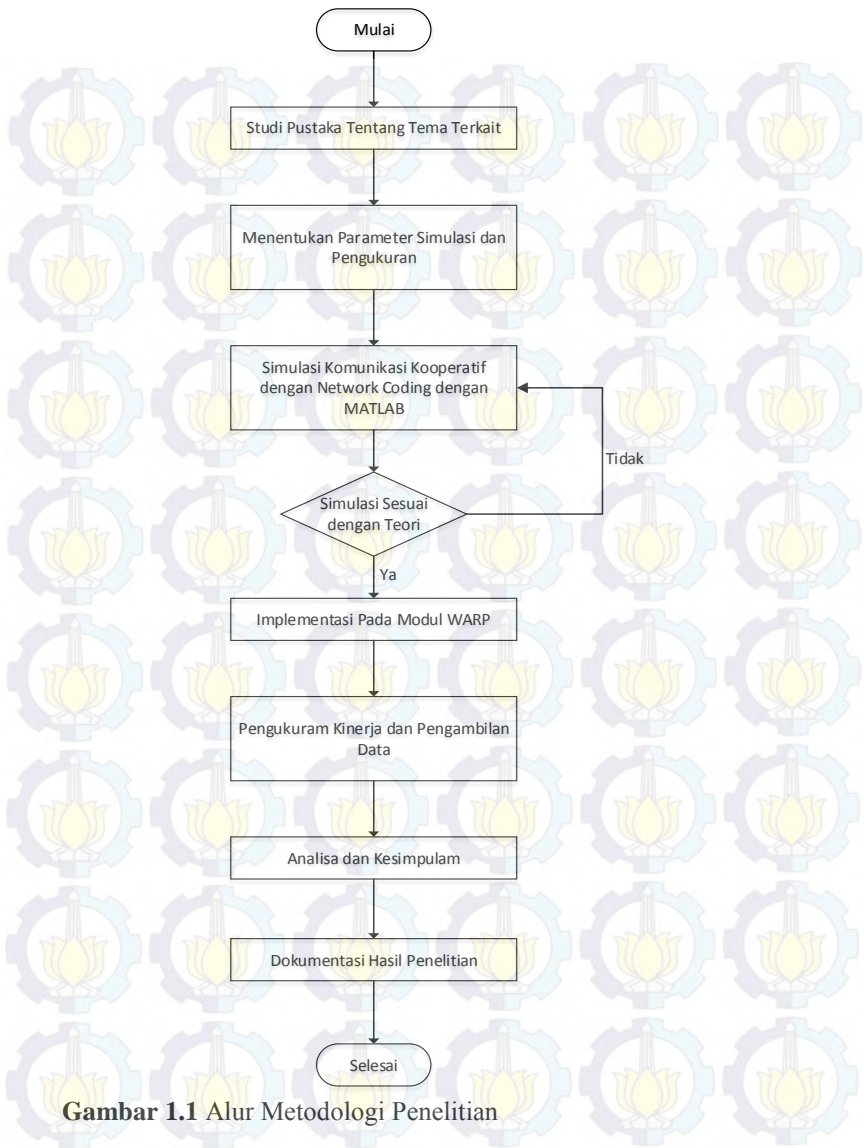
1.4 Tujuan Penelitian

Tujuan yang diharapkan tercapai setelah selesainya tugas akhir ini adalah sebagai berikut

1. Mengetahui cara menghubungkan *software* Matlab dengan *Hardware* WARP.
2. Mengetahui kinerja sistem komunikasi kooperatif apabila ditambahkan dengan *network coding*
3. Mengimplementasikan kinerja sistem komunikasi kooperatif dengan *network coding* pada WARP

1.5 Metodologi Penelitian

Metodologi yang digunakan pada pengerjaan tugas akhir ini dijabarkan dengan diagram alir sebagai berikut:



Gambar 1.1 Alur Metodologi Penelitian

1. Studi Pustaka

Pada tahap ini akan dipelajari secara rinci teori yang mendukung tugas akhir seperti berikut :

- Sistem Komunikasi Nirkabel
- *Network Coding*
- Sistem Komunikasi Kooperatif
- Model dari Sistem Komunikasi Kooperatif
- Simulasi Model pada MATLAB
- Implementasi Sistem pada WARP

2. Menentukan Parameter Simulasi Dan Pengukuran

Sebelum proses pengukuran berjalan, parameter performansi ditentukan terlebih dahulu untuk menunjukkan kualitas performa dari suatu sistem, dirumuskan dan ditentukan metode pengukurannya. Pada tahapan ini akan ditentukan misalnya, band frekuensi, informasi yang dipancarkan, modulasi yang dipakai, jumlah bit yang dibangkitkan. Parameter yang diambil sebagai acuan performa sistem adalah *bit error rate* (BER)

3. Simulasi Komunikasi Kooperatif DF Network Coding pada MATLAB

Proses simulasi dimaksudkan bertujuan sebagai acuan sebelum dilakukan pengukuran. Selain itu untuk memastikan kesesuaian dengan dasar teori.

4. Implementasi pada Modul WARP

Proses implementasi akan diterapkan pada modul WARP dengan kondisi dalam ruangan secara real.

5. Pengukuran dan Pengambilan Data

Pengukuran akan dilakukan dengan variasi jarak 3 – 6 meter yang artinya jarak antara tiap node adalah setengah dari jarak total yang ditentukan. Kemudian akan dilakukan pengukuran untuk mengetahui seberapa baik kinerja sistem komunikasi kooperatif *decode and forward* dengan skema *network coding* ini.

6. Analisa dan Kesimpulan

Analisa dan penarikan kesimpulan akan dilakukan setelah pengukuran dilakukan.

7. Dokumentasi Hasil Penelitian

Tahap akhir dari pengerjaan tugas akhir ini adalah dengan membuat dokumentasi hasil penelitian yakni berupa laporan dan jurnal tugas akhir, pembuatan laporan dan jurnal dilakukan sesuai tahapan ini.

1.6 Sistematika Penulisan

Proses Penelitian yang tersusun dalam tugas akhir dapat sebagai berikut :

- BAB 1, Bagian awal pada laporan ini membahas tentang latar belakang, perumusan masalah, batasan masalah tujuan penelitian sampai manfaat yang dikerjakan pada tugas akhir ini.
- BAB 2, Bagian kedua membahas tentang landasan teori yang digunakan sebagai acuan dalam mengerjakan tugas akhir ini.
- BAB 3, Bagian ketiga membahas tentang perancangan dan implementasi sistem mulai dari pembangkitan data, encoding, modulasi, decoding, proses network coding, combining sampai data diterima pada node tujuan.
- BAB 4, Bagian keempat membahas tentang hasil pengukuran serta akan dilakukan analisa terhadap data yang didapat dari hasil pengukuran.
- BAB 5, Bagian terakhir adalah penarikan kesimpulan dari analisa yang dilakukan pada bab sebelumnya.

1.7 Manfaat Tugas Akhir

Manfaat yang diharapkan tercapai setelah selesainya tugas akhir ini adalah sebagai berikut :

1. Supaya mahasiswa mengetahui kinerja komunikasi kooperatif dengan *network coding* jika diimplementasikan pada WARP.
2. Menambah pengetahuan seputar perangkat WARP dan kegunaannya.



BAB 2

TINJAUAN PUSTAKA

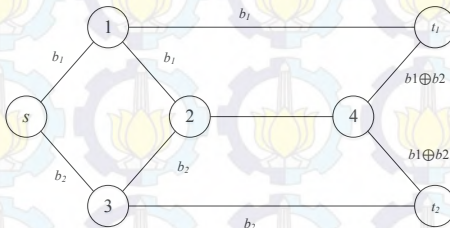
2.1 Sistem Komunikasi Nirkabel

Sistem komunikasi nirkabel merupakan proses transfer informasi baik itu data maupun suara diantara dua atau lebih titik yang tidak terhubung secara fisik. Mekanisme sistem komunikasi nirkabel ini memungkinkan komunikasi jarak jauh dengan daerah yang tidak mungkin atau relatif sulit untuk dihubungkan dengan kabel. Namun permasalahan yang selalu ada dalam sistem komunikasi nirkabel adalah media yang digunakan sangat rentang terjadinya *multipath fading*.

2.2 Network Coding

Network Coding adalah teknik di mana data yang dikirimkan dikodekan dan diterjemahkan untuk meningkatkan *throughput*, mengurangi keterlambatan dan membangun jaringan yang lebih kuat. *Network coding* telah digunakan dalam banyak sistem seperti jaringan nirkabel sensor, jaringan nirkabel, jaringan *multicast video*, (P2P) jaringan *Peer-to-Peer* dan banyak lainnya [6].

Cara kerja *network coding* adalah dengan mengkodekan informasi dengan xor disetiap nodenya, untuk lebih jelasnya dapat dilihat pada gambar 2.1.



Gambar 2.1 Butterfly Network

Dari gambar diatas dapat dilihat bahwa node *source* mengirimkan informasi b_1 dan b_2 ke node 1 dan 3, kemudian pada node 2 dikodekan dengan menggunakan xor, sementara itu node 1 juga mengirimkan informasi b_1 ke node t_1 dan node b_2 juga mengirimkan ke node t_2 , sementara itu node 2 mengirimkan informasi yang sudah dikodekan ke node 4 yaitu $b_1 \oplus b_2$ yang kemudian dikirimkan lagi ke node t_1 dan t_2

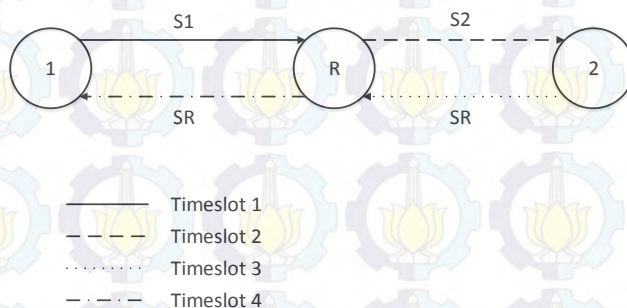
untuk dikodekan dengan xor, sehingga node t_1 akan mendapat informasi b_2 dan node t_2 akan mendapatkan informasi b_1 [7].

Kelebihan dari *network coding* adalah sebagai berikut :

- Meningkatkan *throughput*
- Meningkatkan Kompleksitas
- Meningkatkan Keamanan

2.2.1 Skema Tradisional Tanpa *Network Coding*

Skema Pada skema tradisional tanpa *network coding*, dengan menggunakan 4 *time slot* pada menggunakan 2 *packets*. Pada *time slot* 1, *node* 1 mengirimkan paket S_1 ke *relay* R. Pada *time slot* 2, *Relay* R mengirimkan kembali S_1 ke *node* 2. Pada *time slot* 3, *node* 2 mengirim paket S_2 ke *relay* R dan Pada *time slot* 4, *relay* R mengirimkan kembali informasi S_2 ke *node* 1. Ilustrasi tersebut dapat dilihat pada gambar 2.2 [8]



Gambar 2.2 Skema Tradisional Tanpa *Network Coding*

2.2.2 Skema *Straight Network Coding*

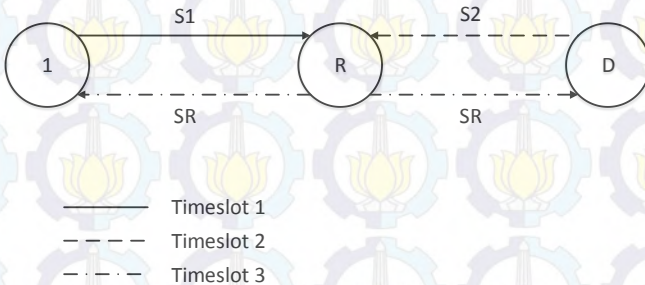
Pada *straightforward network coding*, menggunakan *time slot* 3 sehingga dapat memiliki kemampuan untuk menaikkan *throughput* sebesar 33%. Pada gambar 2.12 menggambarkan ilustrasi SNC menggunakan 3 *time slot* untuk proses pengiriman data informasi. Pada *time slot* 1, *Node* 1 mengirimkan S_1 ke *relay*. Kemudian pada *time slot* 2, *node* 2 mengirimkan informasi S_2 ke *relay*. Setelah menerima informasi dari S_1 dan S_2 , *relay* kemudian membantuk pemetaan *network coding* $S_R = f(S_1, S_2)$. Sedangkan pada S_1, S_2 terdiri dari simbol modulasi dan $f(S_1, S_2)$ merupakan XOR, sebagai 2.3 :

$$S_R = S_1 \oplus S_2 \quad (2.1)$$

Pada time slot ke 3, Relay R membroadcast S_R ke node 1 dan node 2. Pada saat node 1 menerima informasi dari S_R , S_2 mengekstrak informasi dari S_R menggunakan informasi dari S_1 . Sehingga dapat ditulis sebagai berikut

$$S_1 \oplus S_R = S_1 \oplus (S_1 \oplus S_2) = S_2 \quad (2.2)$$

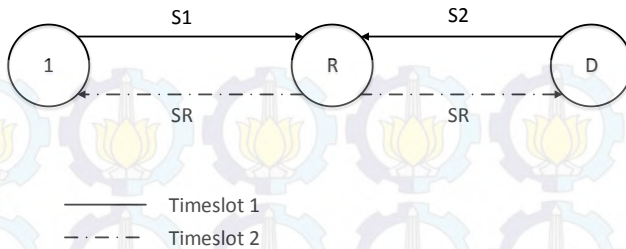
Demikian juga pada *node 2*, mengekstrak informasi S_1 dari $S_2 \oplus S_R$. Proses network coding dilakukan oleh relay setelah melewati proses decoding dari paket – paket node 1 dan 2 pada time slot yang berbeda [8].



Gambar 2.3 Skema *Straight Network Coding*

2.2.3 Skema Physical Network Coding

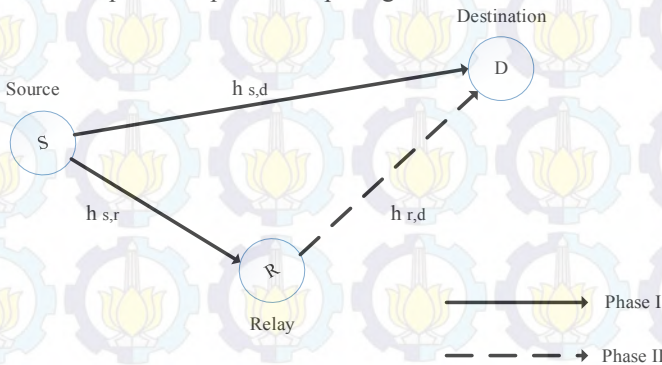
Pada *physical layer network coding* diharapkan dapat mengurangi jumlah time slot, dengan hanya membutuhkan dua *time slot* untuk mengirimkan paket secara bersamaan dan memodifikasi operasi network coding secara alamiah pada gelombang EM. Hal ini memungkinkan node 1 dan node 2 untuk mengirimkan informasi secara bersama-sama sehingga dapat meningkatkan throughput hingga 100 %. Gambar 2.13 menggambarkan ilustrasi skema *physical layer network coding*. Pada *slot* pertama, node 1 dan node 2 mengirimkan informasi pada S_1 dan S_2 secara bersamaan ke *relay*. PNC berbasis gelombang elektromagnetik. Berdasarkan superposisi gelombang EM membawa S_1 dan S_2 , relay R mengkodekan $SR = S_1 \oplus S_2$. Kemudian pada time slot kedua, relay R membroadcast S_R ke node 1 dan node 2.



Gambar 2.4 *Physical Network Coding*

2.3 Sistem Komunikasi Kooperatif

Sistem komunikasi kooperatif merupakan mekanisme yang mengumpulkan sumber daya dari node-node yang terdistribusi yang berada disekitar untuk memaksimalkan kinerja jaringan [9]. Pada sistem komunikasi kooperatif, *source* (S) mengirimkan informasinya ke *relay* (R) dan *destination* (D) yang merupakan *user* lain yang berada didekatnya. Kemudian sinyal yang diterima oleh *relay* akan diolah terlebih dahulu sebelum dikirim ke *destination*. Sinyal yang diterima oleh *destination* berasal dari *source* dan *relay*, selanjutnya sinyal akan mengalami proses *combining* [10]. Gambaran sederhana dari sistem komunikasi kooperatif dapat dilihat pada gambar 2.5.



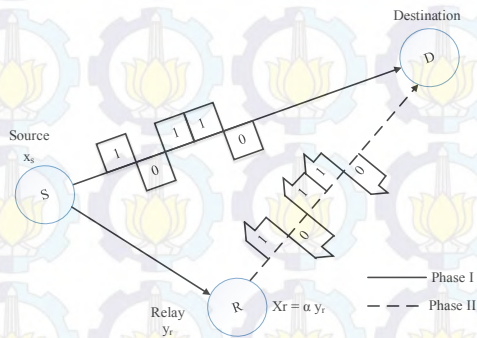
Gambar 2.5 Skema Sistem Komunikasi Kooperatif

2.4 Teknik Forward

Dalam sistem komunikasi kooperatif terdapat beberapa metode *forward* yang diantaranya :

2.4.1 Amplify and Forward

Pada metode *amplify and forward* relay akan menerima sinyal yang terkena noise dari proses transmisi kemudian sinyal dikuatkan dan dikirim ke *destination*. Pada AF, sinyal yang diterima oleh *relay* akan dikuatkan terlebih dahulu sebelum dikirimkan kembali ke *destination*. Metode ini dinamakan dengan metode *fixed protocol* yang mana banyak digunakan ketika waktu pemrosesan yang dimiliki *relay* sedikit sedangkan daya yang tersedia banyak sehingga proses decoder dan encoder harus dikurangi dan diganti dengan penguatan.

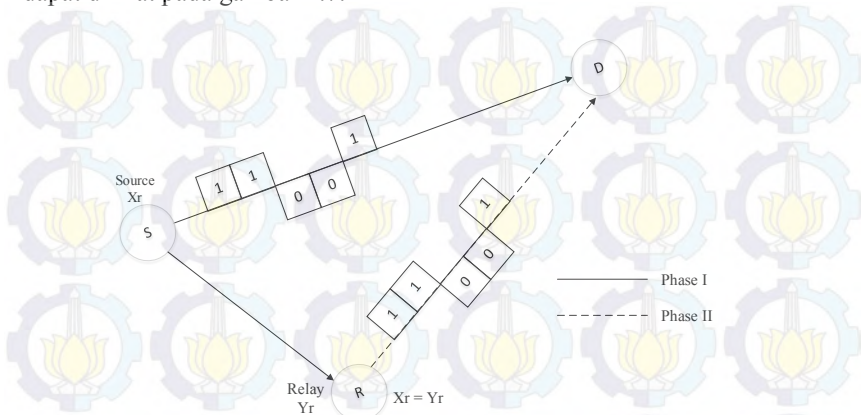


Gambar 2.6 Skema Komunikasi Kooperatif *Amplify and Forward*

2.4.2 Decode and Forward

Pada Metode ini, *relay* akan mendekodekan sinyal yang diterima dari *source* sebelum dikirimkan ke *destination* dan metode ini tidak akan ada penguatan yang terjadi pada sinyal yang akan dikirim tersebut. Pada metode *decode and forward*, sinyal informasi yang diterima *diverlay* akan dikodekan dengan dua cara, pertama dikodekan dengan cara mengkodekan informasi secara keseluruhan dan kedua mengkodekan

sinyal informasi per simbol. Untuk ilustrasi metode *decode and forward* dapat dilihat pada gambar 2.7.



Gambar 2.7 Skema Komunikasi Kooperatif *Decode and Forward*

Secara rinci, dimisalkan $\mathbf{X}_s = [x_s(\mathbf{0}), x_s(\mathbf{1}), \dots, x_s(\mathbf{M} - \mathbf{1})]$. \mathbf{X}_s adalah codeword M simbol yang dikirimkan oleh source dalam phase I dan dimisalkan $E[|x_s[m]|^2] = \mathbf{1}$ untuk semua m . Karena menggunakan media nirkabel maka baik relay dan destination akan menerima noise. Adapun persamaan seperti berikut:

$$y_r(m) = h_{s,r}\sqrt{P_s}x_s(m) + w_r(m) \quad (2.3)$$

$$y_d^{(1)}(m) = h_{s,d}\sqrt{P_s}x_s(m) + w_d(m) \quad (2.4)$$

Dimana $m = 0, 1, \dots, M - 1$

P_s = daya pancar dari source

$h_{s,r}$ = Koefisien kanal dari S ke R

$h_{s,d}$ = Koefisien kanal dari S ke D

$w_r(m) \sim CN(0, \sigma_r^2)$ dan $w_d(m) \sim CN(0, \sigma_d^2)$

Kanal diasumsikan tetap konstan selama proses transmisi dari *codeword* namun variasi secara independen dan identik dari blok ke blok. *Decoding* di relay berhasil jika laju transmisi tidak lebih besar dari pada kapasitas s-r. Untuk menghitung kapasitas kanal maka digunakan persamaan.

$$C_{s,r}(\gamma_{s,r}) = \log_2(1 + \gamma_{s,r}) \text{ bits per kanal} \quad (2.5)$$

$$\text{Dimana : } \gamma_{s,r} = \frac{P_s |h_{s,r}|^2}{\sigma_r^2}$$

Jika laju transmisi informasi adalah R, maka laju transmisi adalah 2R, outage dianggap terjadi jika $2R > C_{s,r}$. Sinyal yang diterima pada destination dalam phase II dirumuskan seperti persamaan

$$y_d^{(2)}(m) = h_{r,d} \sqrt{P_s} x_s(m) + w_d^2(m) \quad (2.6)$$

Dimana $m = 0, 1, \dots, M - 1$

P_d = daya pancar dari source

$h_{r,d}$ = Koefisien kanal dari R ke D

2.5 Teknik Combining

Teknik *combining* yang biasa dipakai pada sistem komunikasi kooperatif adalah :

2.5.1 Selective Combining

SC merupakan teknik *combining* yang paling sederhana dan sering digunakan. Dengan menggunakan beberapa cabang antenna di mana setiap cabang antenna akan menerima sinyal yang telah mengalami *fading*, di sisi *combiner* akan dipilih satu sinyal yang terbaik dari semua sinyal yang diterima. Pemilihan ini didasarkan oleh nilai *Signal to Noise Ratio* (SNR) yang terbesar. Untuk dijelaskan dapat dilihat pada persamaan

$$y_d = x_n, \text{ dimana } n = \arg(\max \gamma_k) \quad (2.7)$$

Dimana $k = 1, 2, 3, \dots, N_r$

Teknik ini memang paling mudah untuk diimplementasikan karena cara kerjanya sangat sederhana. Namun teknik ini bukanlah teknik yang terbaik dan kinerjanya belum optimal. Hal ini dikarenakan pada teknik ini proses deteksi hanya berdasar satu sinyal saja, yaitu sinyal yang memiliki SNR terbesar, dan sinyal yang lain langsung dieliminasi tanpa diproses terlebih dahulu.

2.5.2 Equal Gain Combining

Teknik ini merupakan teknik *combining* yang menjumlahkan semua sinyal informasi yang diterima. Proses combining dapat menggunakan persamaan

$$y_{d(n)} = \sum_{n=1}^K x_n \quad (2.8)$$

K merupakan banyak cabang diversity, setelah semua sinyal yang diterima dijumlahkan, selanjutnya hasil penjumlahan dari sinyal tersebut akan dideteksi untuk mendapatkan estimasi data yang dikirimkan. Teknik ini lebih baik daripada teknik selective combining karena pada teknik ini semua sinyal yang diterima diproses secara simultan untuk mengestimasi data yang dikirimkan.

2.5.3 Maximun Ratio Combining

Teknik ini hampir sama dengan teknik *equal gain combining*, yaitu menjumlahkan semua sinyal informasi yang diterima kemudian hasil penjumlahan tersebut dideteksi untuk mendapatkan estimasi data yang dikirimkan. Perbedaan teknik ini dengan teknik *equal gain combining* yaitu pada teknik MRC setiap sinyal yang diterima akan dikalikan dengan nilai konjugasi dari koefisien kanal yang telah dilalui untuk lebih jelasnya dapat dilihat pada persamaan

$$y_{d(n)} = \sum_{n=1}^K x_n h_n^* \quad (2.9)$$

K merupakan banyak cabang diversity dan h_n^* merupakan konjugasi koefisien kanal. Teknik MRC ini merupakan teknik yang terbaik dari ketiga teknik *combining*. Hal ini dikarenakan pada teknik ini semua sinyal yang diterima diproses secara simultan, selain itu setiap sinyal dikalikan dengan konjugasi koefisien kanal masing-masing sinyal.

2.6 Kode Konvolusi

Kode konvolusi merupakan kode linear yang memiliki struktur tambahan dalam generator matriks sehingga operasi pengkodean dapat dipandang sebagai filter ataupun operasi konvolusi. Dalam prakteknya kode konvolusi banyak diimplementasikan pada *hardware* yang didalam tersedia *encoding* dan *decoding*. Konvolusi *encoder* merupakan sebuah filter digital linear, sistem time invariant dengan urutan kode menjadi

output interleaved dari *output filter*. Kode konvolusi banyak digunakan dalam prakteknya karena memberikan kinerja yang lebih baik dibandingkan dengan kode blok yang sebanding. Kode konvolusi didefinisikan sebuah sistem kode yang memiliki bit informasi masukan harus lebih kecil daripada keluaran bit terkode dan yang paling penting kode konvolusi memiliki memori yang menyebabkan terciptanya sebuah aturan untuk mengkodekan setiap informasi bit masukan menjadi beberapa bit terkode berdasarkan bit-bit informasi masukan sebelumnya [12].

Kode konvolusi (n,k,m) merupakan satu set *codeword* dengan k input, n output dan m tingkat memori. Kode konvolusi ini dihasilkan dengan cara melewati urutan bit informasi melalui sejumlah tingkat *shift register*. Pada umumnya *shift register* terdiri dari N (k bit) tingkat dan m generator polynomial. Data masukan digeser sepanjang k bit *shift register* pada satu kali waktu. Jumlah bit keluaran untuk tiap k bit masukan n bit, dengan *code rate* $R = k/n$.

Parameter utama yang menjadi bagian utama dari *convolutional code* antara lain [13] :

1. *Rate*

Merupakan rasio antara masukan bit informasi dengan keluaran bit terkode dan mempunyai persamaan sebagai berikut :

$$R = k/n \quad (2.10)$$

Keterangan :

R = Laju kode konvolusi

k = Jumlah bit masukan kode konvolusi

n = Jumlah bit keluaran kode konvolusi

2. *Constrain Length*

Jumlah *delay* elemen dalam kode konvolusi yaitu memori dengan masukan bit sekarang pada kode konvolusi atau dapat disebut juga panjang kode dari kode konvolusi. *Constraint length* dapat didefinisikan sebagai berikut :

$$K = m + 1 \quad (2.11)$$

Keterangan :

K = *Constraint length*

m = memori

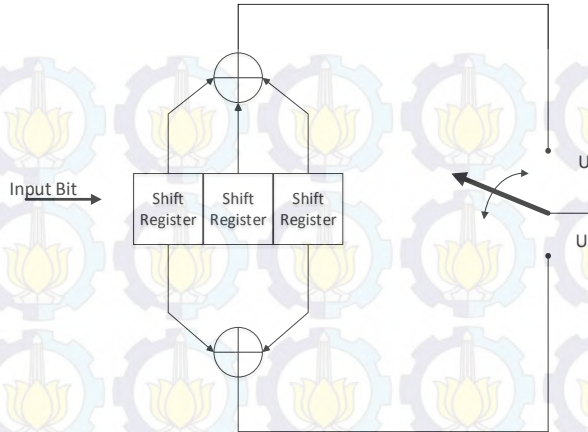
3. *Generator Polynomial*

Generator polynomial sangat dibutuhkan untuk merangkai suatu kode konvolusi berdasarkan jumlah memori yang digunakan dalam suatu kode konvolusi selain itu setiap elemen pada generator polynomial serta jumlah dari generator polynomial mempengaruhi :

- Jumlah output
- Panjang kode konvolusi
- Hubungan antara *shift register* dan modul

Generator polynomial merupakan salah satu metode untuk menggambarkan matriks yang digunakan pada kode konvolusi. Generator polynomial ini biasanya ditulis dalam bentuk oktal. Setiap vektor pada matriks generator memiliki ukuran dimensi Kk dan mengandung garis hubungan pada *encoder* menuju *modulo 2 adder*. Masukan nilai “1” pada posisi I (baris) dari vektor untuk garis hubungan pada *shift register* yang menuju *modulo 2 adder* dengan masukan “0” pada posisi vector jika tidak ada hubungan antara *shift register* dengan *modulo 2 adder* [11].

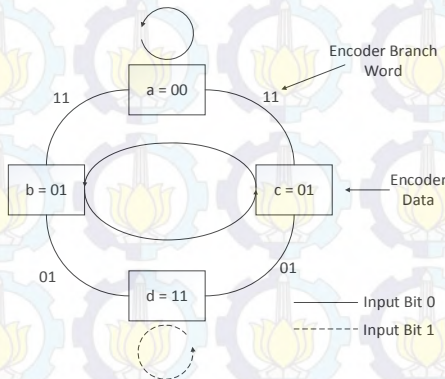
Encoder merupakan bagian kode konvolusi yang melakukan pengkodean data digital (berupa deret biner) yang diterima kode konvolusi sebelum ditransmisikan melalui saluran, sedangkan decoder kode konvolusi lebih rumit yang dimana biasanya menggunakan *Viterbi Algorithm*. Encoder kode konvolusi dengan panjang $K = 3$ dan rate $\frac{1}{2}$ dapat dilihat pada gambar 2.8.



Gambar 2.8 Encoder Kode Konvolusi dengan *Coderate* $\frac{1}{2}$

2.6.1 Diagram State

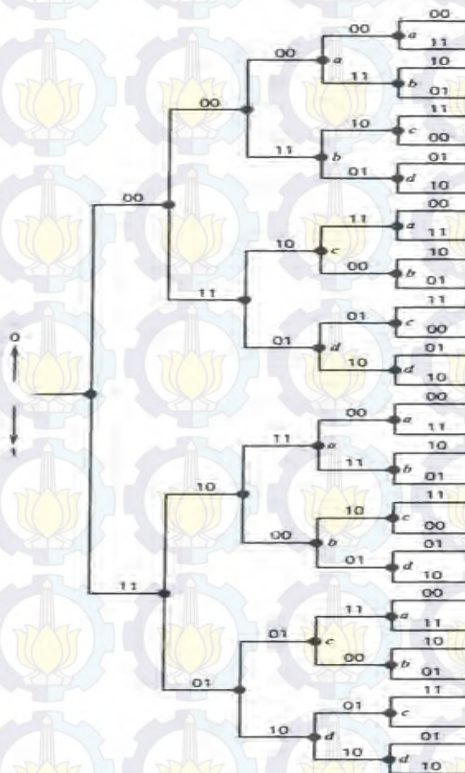
Encoder kode konvolusi dapat direpresentasikan dengan diagram state. State diagram digunakan untuk merepresentasikan proses pengkodean yang berbentuk diagram sederhana. Diagram state akan menunjukkan kemungkinan keadaan transisinya dari satu keadaan ke keadaan lainnya. Diagram state ditunjukkan pada gambar 2.9.



Gambar 2.9 Diagram State Rate = $\frac{1}{2}$ dan $K = 3$ [12]

2.6.2 Diagram Tree

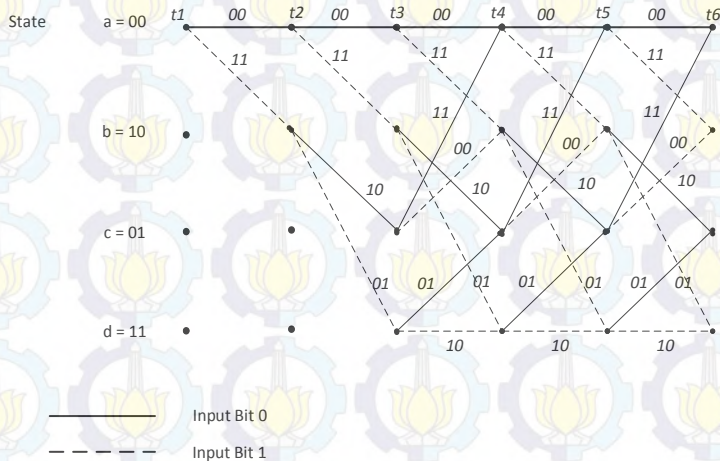
Struktur *encoder* bentuk diagram *tree* dengan cabang-cabangnya menunjukkan variasi keadaan dan keluaran yang mungkin terjadi. Jumlah cabang dalam diagram *tree* digandakan untuk tiap bit masukan yang baru. Dapat dilihat bahwa diagram *tree* berulang setelah cabang dari level kedua, karena setelah level ini hanya terdapat empat sinyal unik yang berbeda. Simpul-simpul ini disebut sebagai *state* dan diberi symbol a,b,c,d. Setiap simpul menghasilkan bit-bit keluaran dan *state* simpul baru, tanpa melihat posisi dari simpul dalam diagram *tree*. Untuk lebih jelas dapat dilihat seperti gambar 2.10.



Gambar 2.10 Diagram Tree Rate = 1/2 dan K = 3 [12]

2.6.3 Diagram Trellis

Diagram trellis merupakan bentuk penyederhanaan dari diagram tree yang merupakan representasi dari keluaran. Ada empat keadaan yang diletakkan sepanjang sumbu vertical, sedangkan transisi diantara state-state direpresentasikan dengan garis vertical dan diagonal yang melintang (menyerupai teralis) yang bergerak ke kanan seiring pertambahan waktu. Contoh untuk diagram trellis dapat dilihat pada gambar 2.11. Diagram ini biasanya digunakan untuk memudahkan proses *decoding*.



Gambar 2.11 Diagram Trellis $\text{Rate} = \frac{1}{2}$ dan $K = 3$ [12]

2.6.4 Algoritma Viterbi

Algoritma Viterbi pertama kali diusulkan sebagai solusi untuk mengkodekan kode konvolusi oleh Andrew J. Viterbi pada tahun 1967. Pada pengembangannya algoritma ini digunakan dalam pemrograman dinamis dengan menggunakan prinsip *maximum likely-hood*. Algoritma Viterbi meminimalkan probabilitas error dengan membandingkan serangkaian kemungkinan *transition site* yang dapat terjadi, dan memutuskan mana yang memiliki probabilitas tertinggi. Algoritma Viterbi memiliki penggunaan yang luas dalam berbagai aplikasi. Penggunaannya tidak hanya dalam bidang telekomunikasi naum juga dalam bidang lainnya misalnya pengenalan tulisan tangan

Prinsip dasar algoritma Viterbi adalah memilih lintasan yang mempunyai node minimum menggunakan sistem *hard decision decoding*. Algoritma Viterbi dapat digambarkan dengan sebuah diagram trellis. Tiap node diagram trellis merepresentasikan suatu state dan memiliki sebuah cabang untuk setiap perpindahan state yang mungkin dalam setiap interval waktu.

2.7 Modulasi Digital

Modulasi digital merupakan suatu proses dimana simbol – simbol digital diubah menjadi bentuk gelombang yang sesuai dengan karakteristik kanal yang akan dilewati. Modulasi ini berfungsi untuk menyamakan karakteristik sinyal dengan karakteristik kanal yang digunakan, untuk mengurangi *noise* dan interferensi, serta mengatasi keterbatasan peralatan.

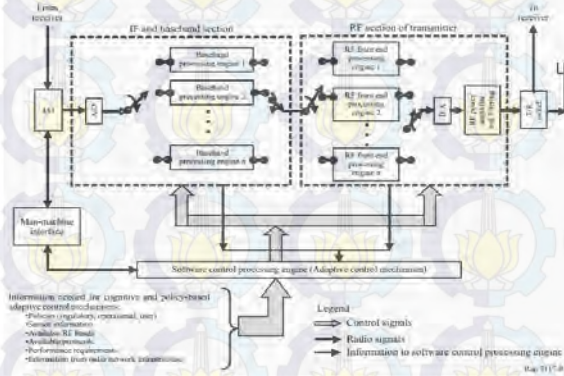
Proses modulasi terdapat pada modulator dan demodulator. Modulator digital berguna untuk memetakan deretan informasi biner menjadi bentuk sinyal gelombang yang nantinya dikirim melalui kanal. Pada modulasi *binary*, modulator digital secara sederhana memetakan digit biner “0” menjadi sinyal gelombang $S_0(t)$ dan digit biner “1” menjadi $S_1(t)$. Demodulator sinyal yang terdapat pada sisi penerima bertugas memproses sinyal yang telah rusak karena proses di kanal, dengan mempresentasikan menjadi beberapa estimasi simbol data sesuai yang dikirimkan.

2.8 Software Defined Radio (SDR)

Software define radio (SDR) atau dapat disebut juga *software radio* adalah suatu sistem komunikasi radio yang dapat disesuaikan pada frekuensi apapun dan menerima modulasi apapun melalui sebuah spektrum frekuensi yang besar. Penggunaan SDR bertujuan untuk mengurangi pergantian *hardware* secara terus-menerus sehingga lebih menghemat biaya. Radio digunakan pada peralatan *wireless communication* dimana *software* mengimplementasikan fungsi *physical layer* dan *link layer*. Hal ini sesuai untuk peralatan *single wireless* yang kemudian diprogram ulang menggunakan modulasi yang berbeda, *coding* dan protokol akses.

Fungsi utama dari *software* adalah dapat diprogram untuk standar yang ada secara fleksibel. Selain itu juga dapat diupdate secara dinamis dengan *software* baru tanpa banyak mengubah *hardware* dan infrastruktur yang sudah ada.

Selain itu SDR juga dapat membangun suatu modem yang kompleks dengan program, *hardware* yang sudah ada dan juga komponen-komponen radio. Adapun arsitektur penyusun yang dimiliki teknologi SDR dapat dilihat pada gambar 2.12.



Gambar 2.12 Arsitektur *Software Define Radio* [14]

Pada dasarnya arsitektur SDR terdiri atas perangkat keras untuk proses baseband, pengolahan sinyal, elemen yang mendukung konversi RF dan teknologi yang dipakai. Komponen yang digunakan untuk *baseband processing engine* bisa menggunakan *General Purpose Processors* (GPP), *Digital Signal Processor* (DSP), dan *Field Programmable Gate Array* (FPGA). Komponen tersebut didukung oleh *Application Programming Interface* (API) dari sistem perangkat lunak radio. Sistem yang dibangun berdasarkan standar arsitektur SDR dari ITU-R M.2117 memiliki keuntungan yaitu lebih ekonomis dari sisi perangkat keras dan perangkat lunaknya.

2.9 Wireless Open-Access Research Platform (WARP)

Wireless Open Access Research Platform (WARP) dikembangkan dilaboratorium CMC, Rice University. WARP menyediakan platform yang scalable dan dapat dikonfigurasi terutama di disain untuk *prototype* algoritma komunikasi nirkabel untuk aplikasi yang berorientasi pada edukasi dan *research*. WARP dapat diprogram dan fleksibel membuatnya mudah untuk mengimplementasikan *protocol physical* dan *network layer* [9]



Gambar 2.13 Modul WARP Versi 2.2 [15]

2.9.1 WARP Board

WARP dapat didefinisikan sebagai salah satu jenis *software defined radio* (SDR). Proses implementasi sistem komunikasi pada WARP dilakukan pada modul WARP sebagai FPGA dan modul Radio sebagai antarmuka antara sistem dengan pengolahan sinyal digital.

Dalam proses pengiriman dan penerimaan sinyal informasi, pengolahan digital dilakukan dengan menggunakan FPGA dan dengan *radio board*. Pada masing-masing *radio board*, akan terdapat dua *buffer* yaitu *Inphase* dan *Quadrature*. Proses pengiriman sinyal dari WARPLab misalnya, akan menggunakan sinyal kompleks.

Komponen utama dari *node* WARP yaitu:

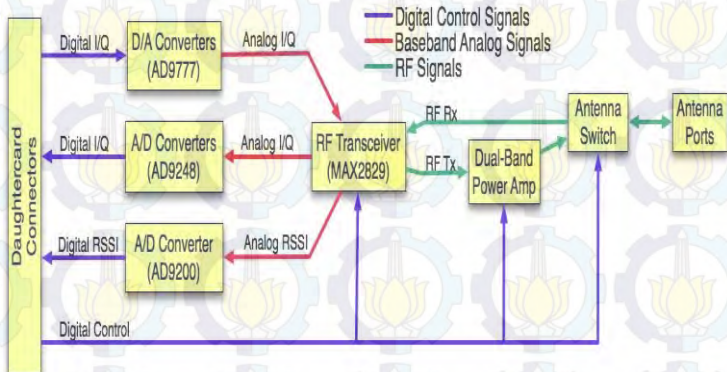
- a. FPGA board
Merupakan inti pemrosesan SDR. Pada implementasi dengan syngen misalnya, High Density Language (HDL) dapat diprogram pada FPGA.
- b. Radio board
Merupakan *interface* dari domain digital ke domain analog. Pada proses konversi dari digital ke analog dan sebaliknya, *radio board* menggunakan ADC (Analog-to-Digital Converter) dan DAC (

Digital-to-Analog Converter). *Radio board* juga melakukan proses modulasi dan demodulasi *baseband*.



Gambar 2.14 *Radio Board* [15]

Radio board pada WARP didesain menggunakan IC MAX2829. Single-chip MAX2829 merupakan IC RF transceiver yang didesain khusus untuk aplikasi OFDM 802.11 WLAN yang dapat bekerja pada frekuensi RF 2.4 GHz- 2.5 GHz dan 4.9 GHz- 5.875 GHz. Untuk arsitektur dari radio board WARP dapat dilihat pada gambar 2.15.



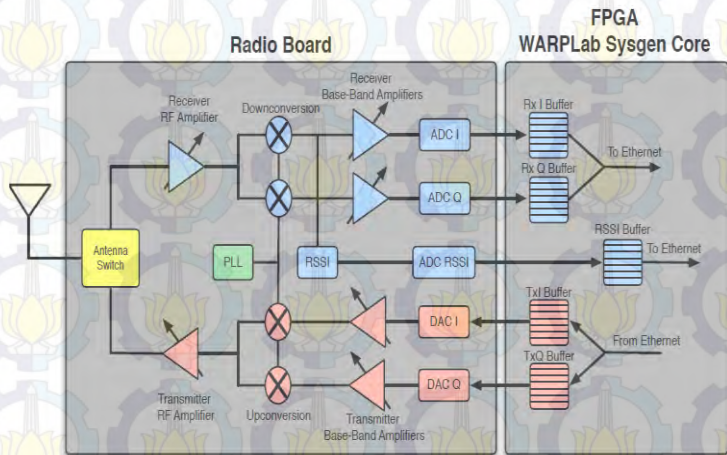
Gambar 2.15 Arsitektur *Radio Board* WARP [15]

Ketika radio board bekerja menjadi transmitter, sinyal digital dari modul WARP akan diubah menjadi analog melalui AD9777 kemudian dipancarkan menggunakan MAX2829. Ketika modul berkerja

sebagai *receiver*, akan terdapat dua sinyal analog yang diproses oleh WARP. Sinyal pertama adalah sinyal informasi I/Q dan kedua adalah sinyal RSSI (*Receive Signal Strength Indicator*).

2.9.2 WARPLab

WARPLab merupakan *framework* untuk prototype *physical layer* (PHY) dengan MATLAB dan *hardware* WARP. Dengan WARPLab, kita dapat secara langsung membangkitkan sinyal dari MATLAB dan dapat ditransmisikan secara *realtime* menggunakan node WARP. Dari gambar 2.16 menunjukan jalur pemancar dan penerima, jika blok pemancar ditandai dengan warna merah dan blok penerima ditandai dengan warna biru.



Gambar 2.16 Arsitektur WARPLab [15]

Blok pada bagian pemancar :

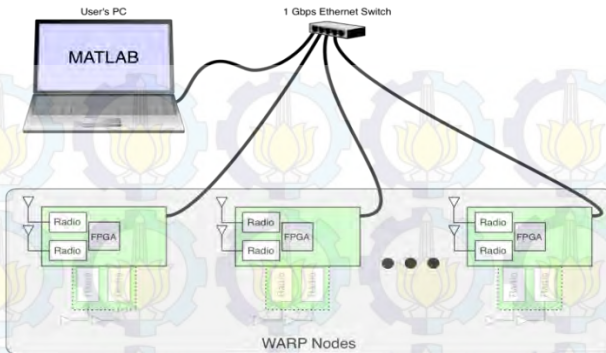
1. Tx I/Q Buffers: Sampel *In-Phase* yang akan ditransmisikan terlebih dahulu disimpan di dalam Tx I Buffer and Sampel *Quadrature* yang akan ditransmisikan terlebih dahulu disimpan di dalam Tx Q Buffer.
2. DAC I/Q: *Digital-to-Analog Converter* untuk sampel *In-Phase/Quadrature*.
3. Tx BB Amplifiers: *Transmitter Base Band Amplifiers*. Pada Tx BB terdapat dua *amplifier*, satu untuk signal I dan satu untuk signal Q. Bagian adalah *variable gain amplifier*, keduanya diset dengan nilai

gain sama yang dapat dimasukkan oleh user menggunakan MATLAB.

4. *Upconversion*: mengkonversi sinyal base band ke sinyal RF. Frekuensi carrier bergantung kepada setting PLL.
5. *Tx RF Amplifiers*: pemancar *RF amplifier*, *variable gain amplifier*, nilai *gain* dapat dimasukkan oleh user menggunakan MATLAB.

Blok pada bagaian penerima:

1. *Rx RF Amplifier*: *Receiver RF amplifier*, *variable gain amplifier*, nilai *gain* dapat dimasukkan oleh user menggunakan MATLAB.
2. *Downconversion*: mengkonvesri signal RF ke *base band In-Phase/Quadrature* signals. Downconversion dari RF bergantung kepada setting PLL.
3. *Rx BB Amplifiers*: *Receiver Base Band Amplifiers*. Ada dua *amplifier*, satu untuk sinyal I dan satu untuk sniyal Q. Bagian ini adalah *variable gain amplifier*, keduanya diset dengan nilai *gain* sama yang dapat dimasukkan oleh user menggunakan MATLAB.
4. *RSSI*: *Received Signal Strength Indicator*. Blok ini mengukur *RSSI*.
5. *ADC I/Q*: *Analog-to-Digital Converters* untuk sample *In-Phase/Quadrature*.
6. *ADC RSSI*: *Analog-to-Digital Converter* untuk mengukur *RSSI*. Data *RSSI* tersedia di 1/4 rate dari data I/Q.
7. *Rx I/Q Buffers*: Menerima sampel *In-Phase* disimpan di *Rx I Buffer* dan Menerima sampel *Quadrature* disimpan di *Rx Q Buffer*.
8. *RSSI Buffer*: data *RSSI* disimpan di dalam buffer ini.
9. Blok *PLL* : merupakan pengaturan *PLL* untuk menentukan frekuensi pembawa yang dapat diatur pada salah satu dari kanal 14 *Wi-Fi* 2,4 GHz band atau 23 saluran di 5 GHz. Untuk mengatur saluran pembawa dapat langsung diatur dari *workspace* Matlab.

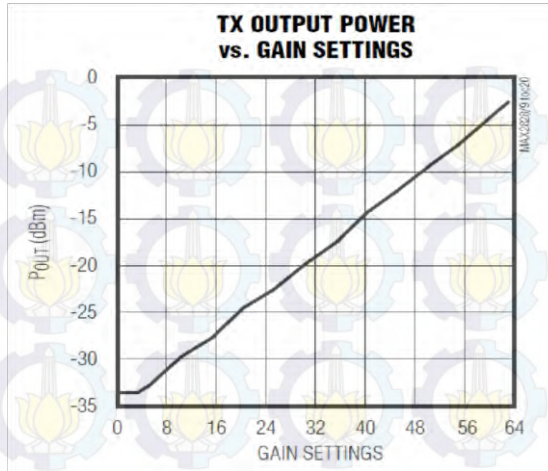


Gambar 2.17 Penggunaan Kerangka Kerja WARPLab [15]

Masing-masing node yang diprogram melalui WARPLab terhubung dengan komputer dengan menggunakan switch Ethernet. Dengan menggunakan sistem ini maka node dapat diprogram untuk mengirim data dan menerima data menggunakan MATLAB.

2.10 Receive Strength Signal Indicator (RSSI)

Pada RSSI (*Receive Signal Strength Indicator*) ini dapat diketahui level tegangan tertentu pada penerima. Nilai level tegangan ini dapat kita ketahui berdasarkan informasi dari datasheet. Pada gambar akan terlihat karakteristik sinyal analog RSSI pertama dihasilkan oleh MAX2829. Untuk menentukan daya pancar pada pengirim pengaturan nilai gain RF dilakukan. Pada gambar 2.18 diperlihatkan grafik konversi untuk range gain RF terhadap nilai daya pancar dalam dBm.



Gambar 2.18 TX Output vs Gain Settings pada MAX2829 [16]

Berdasarkan referensi dari data sheet MAX2829 [16] karakteristik RF gain memiliki kaitan sebagai berikut :

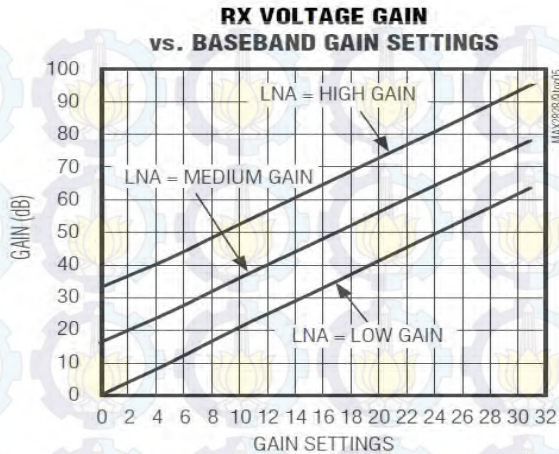
- a. Nilai gain linear pada range 4 hingga 63.
- b. Daya output dinyatakan dalam persamaan terhadap gain setting sebagai berikut :

$$P_{out} = 0.5252x - 36.1017 \quad (2.12)$$

Dimana x adalah gain setting dengan range 4 hingga 63

Sedangkan pada *receiver baseband gain*, terdapat 32 level pengaturan [0:31] berdasarkan referensi pada data sheet MAX2829 [Maxim] grafik level pengaturan *baseband* terhadap nilai gain penerima ditunjukkan pada gambar

Untuk pengaturan RF gain di penerima, node WARP memiliki tiga level LNA, masing – masing sebagai *Low Gain*, *Medium Gain* dan *High Gain*. Besar penguatan LNA sekitar 0 dB untuk *low gain*, 15 dB untuk *medium gain* dan 30 dB untuk *high gain*. Besarnya penguatan baik RF Gain maupun BB gain pada Rx mengacu pada gambar 2.19 yang memperlihatkan peningkatan gain untuk setiap perubahan level LNA dan juga Baseband Gain.



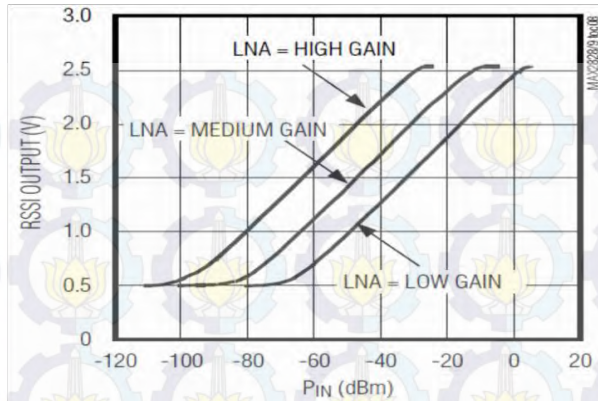
Gambar 2.19 Gain Rx Baseband pada MAX2829 [16]

RSSI juga digunakan untuk mendapatkan nilai daya terima. Pembacaan RSSI menggunakan 10-bit linear dalam dB yang dapat dikonversi menjadi daya terima dalam dBm. RSSI diukur setelah RF amplifier, sehingga konversi nilai RSSI menjadi daya terima bergantung pada pengaturan Gain RX_RF [Maxim].

- a. Untuk High Gain (RX_RF = 3), nilai RSSI = 0 adalah -100 dBm dan RSSI = 1023 adalah -30 dBm.
- b. Untuk High Gain (RX_RF = 2), nilai RSSI = 0 adalah -85 dBm dan RSSI = 1023 adalah -15 dBm.
- c. Untuk High Gain (RX_RF = 1), nilai RSSI = 0 adalah -70 dBm dan RSSI = 1023 adalah 0 dBm.

Dalam hal ini, persamaan konversi dari RSSI kedalam nilai dBm daya terima yang digunakan adalah :

$$P_{rx}(dBm) = \left(\frac{70}{1023}\right) \times RSSI - 70 - (R_x Gain_{RF} - 1) \times 15 \quad (2.13)$$



Gambar 2.20 Grafik RSSI vs Daya Input pada MAX2829 [16]

Dapat dilihat pada grafik yang ditunjukkan Gambar 2.20 bahwa daya input yang terukur oleh receiver akan diubah menjadi tegangan pada range 0.5 – 2.5 Volt. Berdasarkan diagram blok pada Gambar 2.19 sinyal analog akan diubah menjadi sinyal digital melalui A/D converters (AD9200).



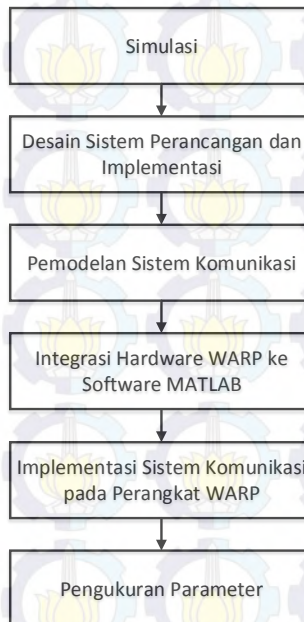
BAB 3

PERANCANGAN DAN IMPLEMENTASI

Pada penelitian ini terdapat beberapa langkah yang dilakukan antara lain simulasi sistem komunikasi, perancangan sistem komunikasi yang terdiri dari penentuan parameter, mengintegrasikan perangkat lunak MATLAB ke perangkat keras WARP serta menentukan scenario pengukuran.

3.1 Alur Perancangan dan Implementasi Sistem

Sebelum meralisasikan perancangan dan implementasi, terlebih dahulu dibuat alur diagram perancangan dan implementasi untuk mengidentifikasi langkah yang dilakukan dalam penelitian ini. Alur perancangan dan implementasi sistem dapat dilihat pada gambar 3.1.



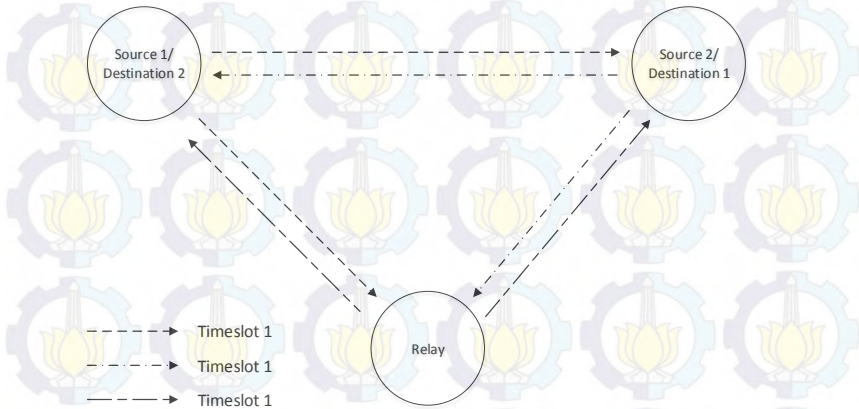
Gambar 3.1 Diagram Alur Perancangan dan Implementasi Sistem

3.2 Simulasi Sistem Komunikasi

Sebelum melakukan implementasi, akan dilakukan simulasi sistem untuk mengetahui apakah sistem yang dibuat akan berjalan pada simulasi.

3.2.1 Skema Simulasi Sistem

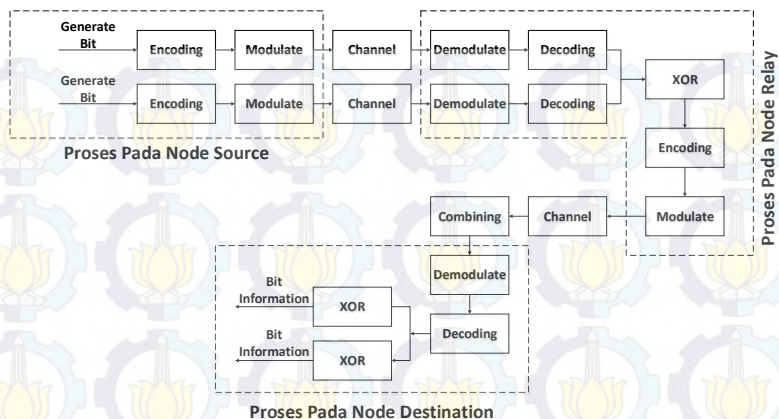
Berikut adalah skema sistem komunikasi yang akan disimulasikan yang terdiri dari tiga *node* dan terdiri dari tiga *timeslot*. Untuk ilustrasi skema *network coding* pada sistem komunikasi kooperatif DF dapat dilihat pada gambar 3.2.



Gambar 3.2 Skema Simulasi Skema *Network Coding* pada Sistem Komunikasi Kooperatif DF

3.2.2 Diagram Blok Simulasi Sistem

Berikut adalah blok diagram simulasi sistem komunikasi, blok diagram sistem komunikasi ini terdiri dari tiga bagian yaitu *node*, *relay*, dan *destination*. Diagram blok simulasi sistem komunikasi dapat dilihat pada gambar 3.3.

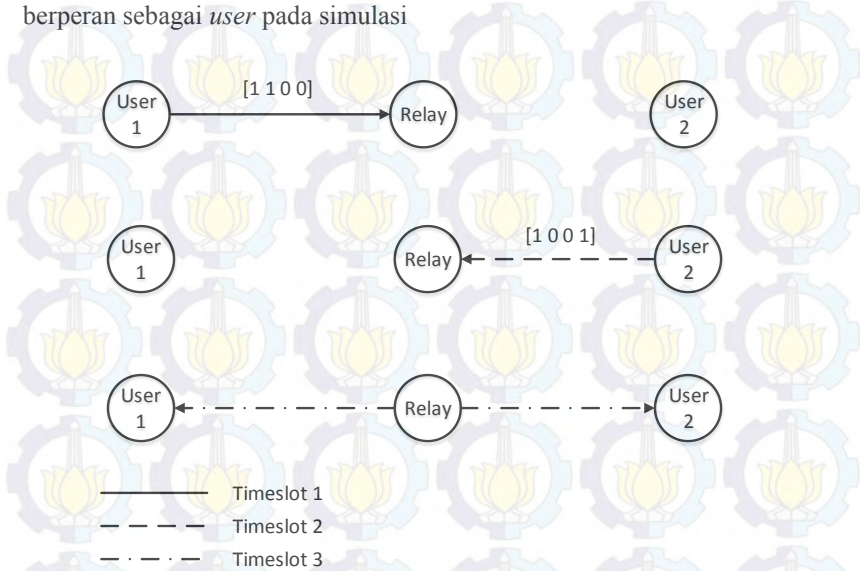


Gambar 3.3 Diagram Blok Simulasi Skema *Network Coding* pada Sistem Komunikasi Kooperatif DF

Untuk penjelasan blok diagram sistem diatas adalah, data dibangkitkan dari *source*, selanjutnya dilakukan proses *encoding* dengan menggunakan kode konvolusi. Setelah itu data di *mapping* menggunakan modulasi *Quadrature Phase Shift Keying* (QPSK). Pada bagian *relay* data yang telah di *mapping* akan di *demapping* dengan menggunakan demodulator QPSK, kemudian akan di *decoding* untuk mendapatkan data seperti saat dikirim, setelah dilakukan proses *decoding* akan dilakukan proses XOR yang merupakan inti dari proses *network coding* kemudian akan dilakukan proses yang sama seperti saat data dikirim. Pada blok penerima data yang diterima proses yang dilakukan akan sama seperti pada saat data masuk pada blok *relay* perbedaannya adalah pada saat melakukan *combining*, pada penelitian ini teknik *combining* yang digunakan adalah *selective combining* yaitu memilih sinyal terbesar antara sinyal yang berasal dari *relay* dan *source*. Setelah dilakukan proses *combining* akan dilakukan proses *demapping* menggunakan demodulator QPSK, kemudian data akan di *decoding* dan selanjutnya di XOR dengan bit informasi pada masing-masing *source*. Pada simulasi ini kanal yang digunakan adalah *Additive White Gaussian Noise* (AWGN). Untuk ilustrasi proses *network coding* pada simulasi sistem komunikasi dapat dilihat pada gambar 3.4 – 3.9.

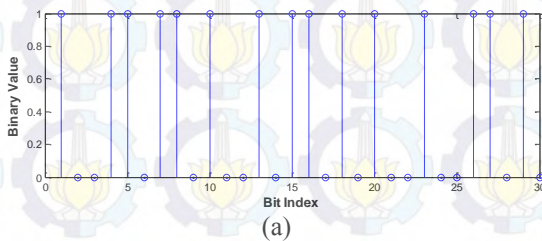
- **Proses pada Pengirim**

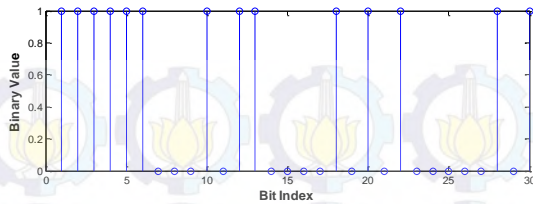
Gambar 3.4 adalah proses pengiriman data dari node *source* yang berperan sebagai *user* pada simulasi



Gambar 3.4 Proses Pengiriman Data pada *Node Source*

Pada *node source* 1 dan 2, akan dikirim bit, sebagai ilustrasi gambar 3.3 memperlihatkan pada saat bit dikirim dari *node source* 1 dan 2, data yang dikirim berupa bilangan biner yang selanjutnya akan dikirimkan ke *relay* dari *node source* 1 dan 2. Data yang dibangkitkan pada Matlab berupa bilangan biner acak dengan fungsi “randi”. Hasil data yang dibangkitkan pada Matlab dapat dilihat pada gambar 3.5.



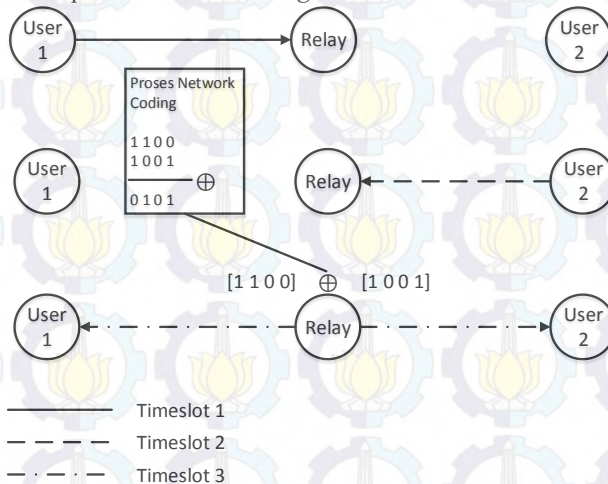


(b)

Gambar 3.5 (a) Bit yang Dikirim dari *Source 1*, (b) Bit yang Dikirim dari *Source 2*

- **Proses pada Relay**

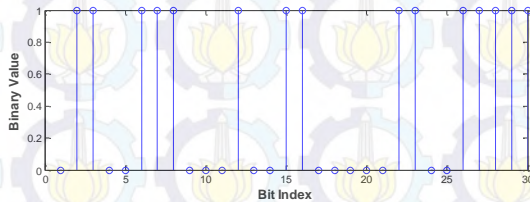
Gambar 3.6 adalah proses saat data sampai pada *node relay*, yang berperan saat proses *network coding*



Gambar 3.6 Proses *Network Coding* pada *Node Relay*

Proses *network coding* terjadi pada *node relay*, yakni dengan menggunakan logika XOR untuk mengkodekan informasi dari *node 1* dan *2*, sebagai ilustrasi dapat dilihat pada gambar 3.4, *node relay* mengkodekan bit informasi dari *node 1* dan *2*, yang selanjutnya akan dilakukan XOR sebagai proses *network coding* untuk selanjutnya informasi akan dikirim ke masing – masing *node destination*. Proses *network coding* pada Matlab

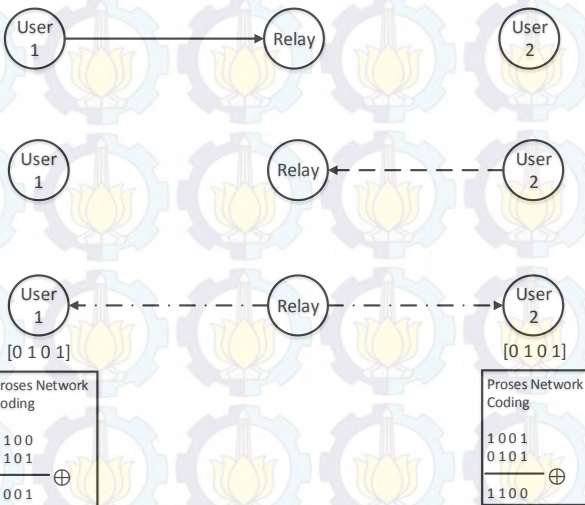
menggunakan fungsi xor, dengan *syntax* xor(data 1,data 2). Hasil XOR pada *relay* dapat dilihat pada gambar 3.7.



Gambar 3.7 Data yang Diterima pada Relay Setelah Proses XOR

- **Proses pada Penerima**

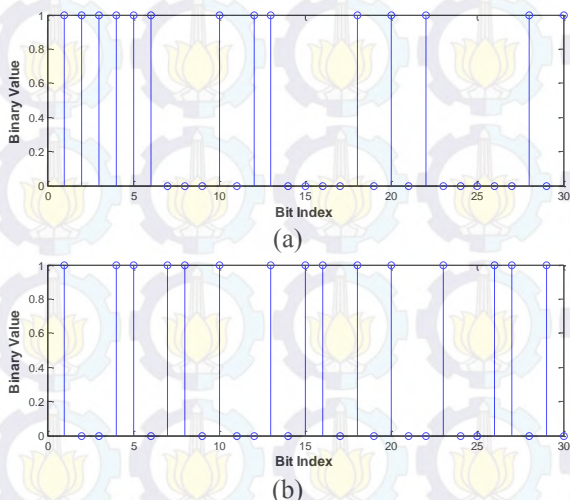
Gambar 3.8 adalah proses *network coding* pada saat informasi sampai di *node destination*.



Gambar 3.8 Proses *Network Coding* pada *Node Destination*

Selain pada *node relay*, proses *network coding* juga terjadi pada *node destination*, tujuannya adalah supaya informasi dari *node 1* terkirim ke *node 2*. Gambar 3.5 memperlihatkan ilustrasi proses *network coding* pada *node destination*, setelah informasi dikirim oleh *node relay* ke *node destination*, informasi yang telah sampai akan di *decoding* dengan

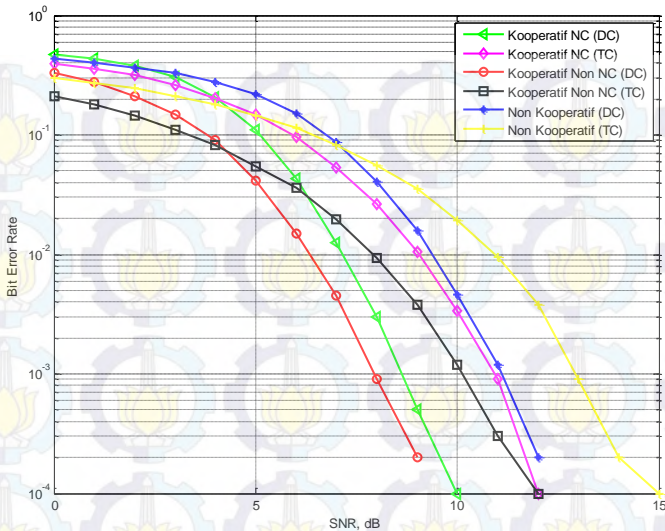
menggunakan XOR antara informasi yang dikirim oleh *relay* dengan informasi asal yang ada pada *node* 1 dan 2, sehingga informasi yang dikirim dari *node* 1 akan terkirim ke *node* 2 begitupun sebaliknya. Sama seperti pada *relay*, proses *network coding* di penerima pada Matlab menggunakan fungsi “xor”. Hasil XOR pada penerima dapat dilihat pada gambar 3.9.



Gambar 3.9 (a) Data yang Diterima *Node Destination 2*, (b) Data yang Diterima *Node Destination 1*

3.2.3 Hasil Simulasi Sistem Komunikasi

Untuk simulasi hasil kinerja *network coding* pada komunikasi kooperatif dapat dilihat pada gambar 3.10.

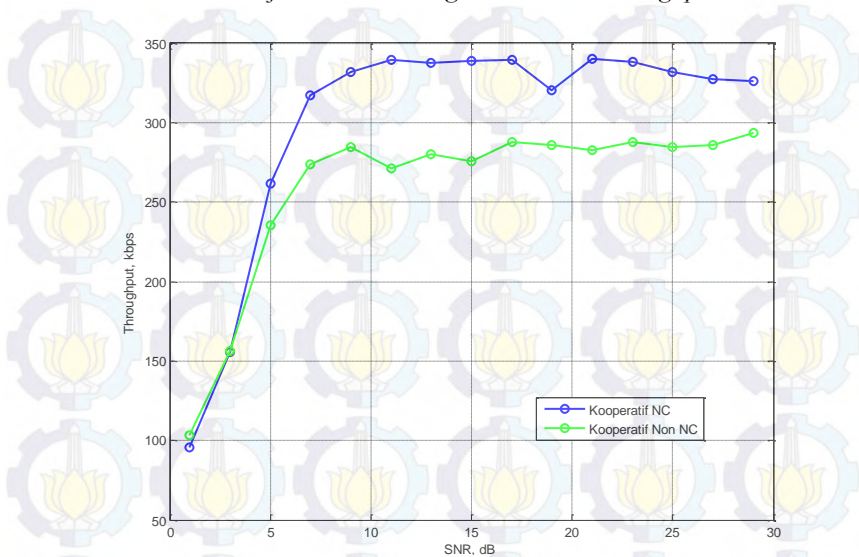


Gambar 3.10 Hasil Simulasi Kinerja *Bit Error Rate* Sistem Komunikasi Kooperatif DF dengan *Network Coding*

Keterangan gambar : TC (Tanpa Coding)
DC (Dengan Coding)

Seperti yang dijelaskan sebelumnya, komunikasi kooperatif dan *network coding* adalah sistem komunikasi yang menggunakan *relay*, untuk pembandingan, sistem komunikasi tersebut akan dibandingkan dengan komunikasi dengan jalur langsung tanpa melewati *relay* (*source – destination*). Dari grafik diatas dapat dilihat bahwa komunikasi kooperatif dan *network coding* mempunyai kinerja lebih baik dibandingkan dengan komunikasi jalur langsung, hal ini dikarenakan adanya *relay* yang berfungsi sebagai *forward* yang berarti setelah sampai pada *relay* bit akan dikembalikan seperti pada sisi pengirim sebelum dikirimkan kembali ke *destination*. Selanjutnya untuk pembandingan, sistem komunikasi kooperatif dan *network coding* maupun jalur langsung yang telah dikodekan performanya lebih baik dibanding yang tidak dikodekan, misalnya pada komunikasi koopeatif *network coding* yang memakai *coding* BER bernilai 0 dicapai pada saat SNR bernilai 10 dB, sedangkan tanpa *coding* BER bernilai 0 dicapai pada saat SNR bernilai 15 dB. Salah

satu keunggulan *network coding* adalah *throughput*, gambar 3.11 adalah hasil simulasi kinerja *network coding* berdasarkan *throughput*.

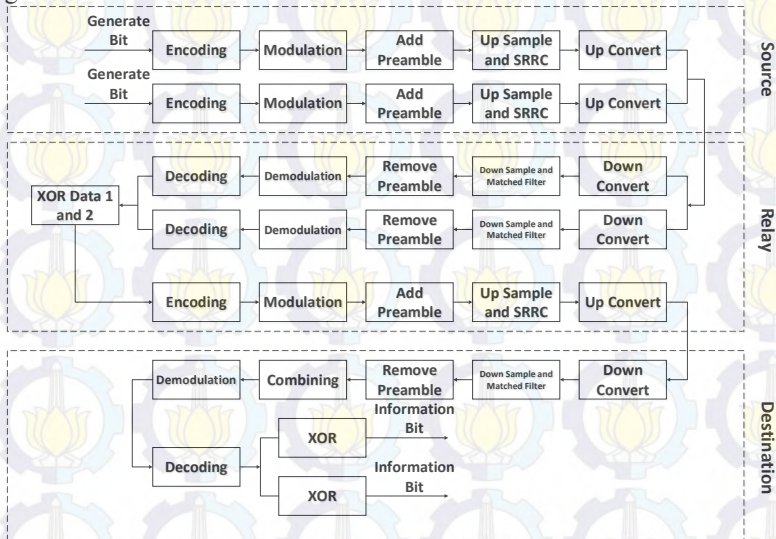


Gambar 3.11 Hasil Simulasi Kinerja *Throughput Network Coding* pada Sistem Komunikasi Kooperatif DF

Dari gambar diatas dapat dilihat bahwa komunikasi kooperatif dengan *network coding* lebih unggul daripada komunikasi kooperatif konvensional dalam hal *throughput*. Misalnya pada saat SNR 5dB, komunikasi kooperatif dengan *network coding* dapat mencapai *bitrate* 260kbps, sedangkan komunikasi kooperatif biasa hanya bias mencapai 220 kbps, *throughput* maksimal yang bias dicapai komunikasi kooperatif konvensional hanya sampai sekitar 260 kbps, sedangkan kooperatif *network coding* bias mencapai hingga sekitar 340 kbps.

3.3 Blok Diagram Perancangan Sistem

Desain sistem terdiri atas *source*, *relay*, dan *destination* serta terbagi menjadi 3 *timeslot*. Untuk blok keseluruhan dapat dilihat pada gambar 3.12.



Gambar 3.12 Diagram Blok Perancangan Skema *Network Coding* pada Sistem Komunikasi Kooperatif DF

Secara umum blok diagram sistem komunikasi kooperatif dengan skema *network coding* ini terdiri dari *source*, *relay*, dan *destination*. Transmisi pada sistem terbagi menjadi dua jalur, yaitu jalur langsung (*source to destination*) dan jalur multihop (*source-relay-destination*). Pada jalur multihop setelah informasi 1 dan 2 sampai di *relay*, informasi tersebut akan dikombinasikan dengan operasi XOR, kemudian informasi tersebut akan ditransmisikan kembali ke tujuan, setelah sampai tujuan informasi tersebut akan diolah dengan menggunakan teknik *selective combining* lalu didemodulasi dan decoder setelah itu informasi yang telah dikombinasikan akan diolah dengan operasi XOR dengan informasi awal supaya mendapatkan data akhir.

3.3.1 Parameter Sistem Komunikasi

Pemodelan sistem komunikasi ini menggunakan beberapa parameter. Berikut adalah beberapa parameter yang digunakan. Parameter yang digunakan pada tugas akhir ini dapat dilihat pada tabel 3.1.

Tabel 3.1 Parameter Sistem Komunikasi

No	Parameter	Nilai
1	Modulasi	<i>Quadrature Phase Shift Keying (QPSK)</i>
2	Filter	<i>Squared Root Raised Cosine (SRRC)</i>
3	Filter order	6
4	<i>Rolloff factor</i>	0.9
5	<i>Preamble</i>	19
6	<i>Sample</i>	8
7	<i>Up converter dan down converter</i>	5 MHz
8	Frekuensi sampling	40 MHz
9	<i>Carrier channel</i>	[0:14]
10	<i>Transmitter baseband gain</i>	[0:3]
11	<i>Transmitter RF gain</i>	[0:63]
12	<i>Receiver basedband gain</i>	[0:31]

3.3.2 Proses pada Pengirim

Blok pengirim yang dimaksud pada gambar 3.1 adalah *source* dan *relay*. Berikut adalah langkah pada blok pengirim.

1. *Generate Bit*

Pada langkah ini bit informasi dibangkitkan secara acak, bit informasi dibangkitkan dalam bentuk biner. dengan fungsi "*randi*" pada Matlab. Setiap bit yang dibangkitkan saling bebas antara satu dengan yang lain dan memungkinkan terjadinya pengulangan.

2. Convolutional Encoder

Bit informasi yang telah dibangkitkan dikodekan dengan kode konvolusi yang terdapat pada Matlab, kode konvolusi pada Matlab dapat digunakan dengan fungsi "*convenc*". Bit informasi dikodekan sesuai dengan *code rate* yang dipakai. Pengkodean dengan metode kode konvolusi menggunakan trellis, dimana trellis merupakan parameter dari *encoder* atau *decoder* yang menunjukkan nilai *constraint length* K , *input* k , dan *output* n .

3. Proses Modulasi

Hasil dari pengkodean kanal selanjutnya akan dimodulasi, modulasi yang digunakan adalah QPSK, yang berarti setiap satu symbol mengandung dua bit informasi.

4. Penambahan Preamble

Preamble disebut juga sebagai *sync word* atau *sync character*, *preamble* dimaksudkan untuk sinkronisasi symbol yang telah ditetapkan dan diketahui oleh sistem yang diletakan pada bagian awal atau akhir deretan informasi.

5. Up Sample dan Filter SRRC

Upsample bertujuan untuk meningkatkan *sampling rate* dari data yang dikirim sesuai dengan jumlah *sample* yang digunakan, sehingga diperoleh sinyal baru dari hasil *sampling*. Hasil *sampling* kemudian akan di filter menggunakan filter *Square Root Raised Cosine* (SRRC).

6. Up Convert

Upconvert merupakan konversi ke frekuensi yang lebih tinggi dengan menggeser frekuensi sinyal tengah *baseband* dilakukan untuk menghindari pelemahan dari radio DC saat mengirim sinyal.

3.3.3 Proses Transmisi pada WARP

Pada proses transmisi sinyal dengan modul WARP, sample sinyal hasil *upconverter* dikirimkan ke *buffer* WARP melalui *switch ethernet* untuk diubah menjadi sinyal analog dengan menggunakan DAC yang terdapat pada *daughtercard radio*. WARP akan mengirim sinyal menggunakan kanal *real* dari *node transmitter* ke *node receiver*. Karena pada tugas akhir ini menggunakan tiga node (*source – relay – destination*) dengan dua node berperan sebagai *source* dan *destination* maka perlu

dilakukan proses inialisasi pada soket yang digunakan dalam hal ini ada tiga node yang akan digunakan, maka soket yang dibutuhkan ada empat dengan satu soket digunakan untuk sinkronisasi. Ada beberapa parameter yang diatur dalam proses transmisi pada modul WARP, parameter dapat dilihat pada tabel 3.1.

Seperti yang telah dijelaskan diatas bahwa pada tugas akhir ini menggunakan tiga *node*, langkah pertama proses pengiriman pada *timeslot* pertama yaitu node 1 (*source 1*) akan mengirimkan sinyal ke node 2 (*relay*) dan node 3 (*destination 1*), selanjutnya pada *timeslot* kedua node 3 (*source 2*) akan mengirimkan sinyal ke node 2 (*relay*) dan node 1 (*destination 2*). Setelah proses tersebut, selanjutnya *timeslot* ketiga node 2 (*relay*) akan mengirimkan sinyal ke node 1 (*destination 2*) dan node 2 (*destination 1*).

3.3.4 Proses pada Penerima

Blok penerima yang dimaksud pada gambar 3.12 adalah *destination* dan *relay*. Proses pada penerima adalah kebalikan dari proses pada pengirim, berikut adalah langkahnya :

1. *Downconvert*

Setelah bit diterima akan dilakukan proses *downconvert*, dimana pada proses ini sinyal akan dikalikan dengan konjugat dari *upconvert*.

2. *Matched Filter*

Pada proses ini, frekuensi tinggi yang dihasilkan dari proses *downconvert* akan dihilangkan agar sesuai dengan bentuk pulsa *raised cosine* disisi pengirim.

3. *Correlator* dan *Downsample*

Sebelum dilakukan proses *downsample* dilakukan proses sinkronisasi *preamble* yaitu dengan mengkorelasikan *sample preamble* sebelumnya dengan *sample* sinyal yang diterima sehingga akan mendapatkan korelasi tertinggi antara *sample* tersebut. Sinyal di *downsample* sesuai dengan *upsample*.

4. *Combining*

Proses selanjutnya dalah mendeteksi letak *preamble*, hal ini dilakukan dengan referensi matrix yang telah di *generate* pada sisi penerima. Pada sistem komunikasi kooperatif aka nada dua sinyal yang

diterima oleh node *destination* yaitu sinyal dari *source – destination* dan *source – relay – destination*, setelah sampai ke penerima akan dilakukan proses *combining*. Pada tugas akhir ini, digunakan *selective combining* yaitu dengan memilih sinyal yang terbaik.

5. Proses Demodulasi

Demodulasi adalah proses pengembalian dari symbol menjadi bit kembali.

6. *Viterbi Decoder*

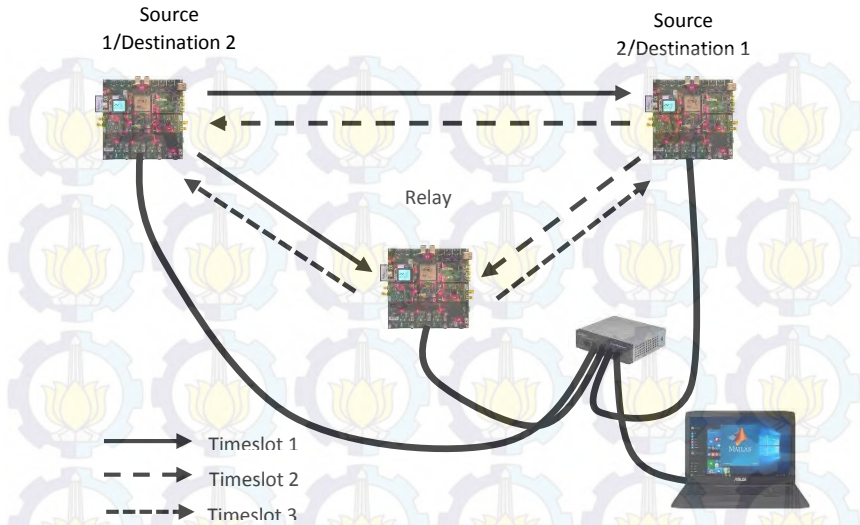
Pada proses ini, hasil *mapping* yang telah didemodulasi menjadi bit kembali akan di *decode* menggunakan Algoritma *Viterbi*, pada Matlab proses ini digunakan dengan fungsi “*vitdec*”. Tujuan proses ini adalah mengembalikan jumlah bit yang telah di *encode*.

7. Proses *Network Coding*

Proses ini bertujuan untuk mendapatkan informasi akhir yang dikirim, pada proses ini informasi berupa bilangan biner yang telah di *encode* akan di XOR untuk mendapatkan informasi akhir.

3.4 Integrasi Laptop/PC ke Perangkat WARP

Pada penelitian ini digunakan tiga node dengan konfigurasi node 1 dan 3 sebagai *source* dan *destination* sedangkan node 2 dikonfigurasi sebagai *relay*. Proses integrasi WARP dengan PC/Laptop menggunakan *switch Ethernet* dan kabel LAN seperti dapat dilihat pada gambar 3.13.



Gambar 3.13 Integrasi PC/Laptop dengan Modul WARP

3.5 Skenario Pengukuran

Tugas akhir ini menggunakan tiga buah modul WARP, dengan masing masing modul berfungsi sebagai *source*, *relay*, dan *destination*. Pengukuran ini dilakukan di *indoor* dengan variasi jarak yang berbeda antara *source* – *destination*. Seperti yang dijelaskan pada bab 1, tujuan dari tugas akhir ini adalah mengimplementasikan sistem komunikasi kooperatif dengan skema *network coding* menggunakan WARP, dengan *network coding* sebagai poin utamanya. Adapaun parameter yang diteliti adalah *bit error rate* (BER) dan *Throughput*.

Dalam sistem komunikasi kooperatif terdapat dua jalur komunikasi, yaitu komunikasi jalur langsung (*source* – *destination*) dan jalur multihop (*source* – *relay* – *destination*). Sedangkan pada sistem *network coding* terdapat sistem XOR yang prosesnya terjadi pada *relay* dan *destination*. Pada timeslot pertama *source* 1 mengirim informasi berupa bit acak ke *relay* dan *destination* 1 (*source* 2). Kemudian pada *timeslot* kedua *source* 2 mengirimkan informasi ke *relay* dan *destination* 2 (*source* 1). Setelah kedua informasi tersebut sampai ke *relay* akan dilakukan proses *network coding* kemudian informasi tersebut di *broadcast* ke *destination* 1 dan 2. Setelah sampai *destination*, informasi

tersebut akan memasuki proses *combining* dengan menggunakan *selective combining* dan setelah itu akan dilakukan proses *network decoding* untuk mendapatkan informasi yang dikirimkan.

Pada tugas akhir ini, pengukuran akan dilakukan dengan dua kondisi, yaitu *indoor* dan *outdoor* dengan variasi jarak yang berbeda beda.

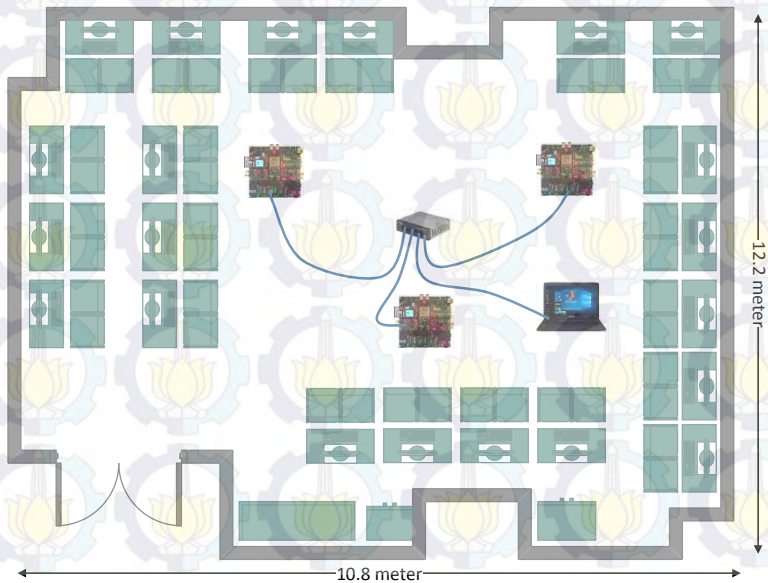
3.5.1 Pengukuran Indoor

Pengukuran *indoor* dilakukan di ruang AJ403 Variasi jarak yang ditentukan pada pengukuran *indoor* dapat dilihat pada tabel 3.2

Tabel 3.2 Variasi Jarak Pengukuran Indoor (AJ403)

Source - Destination	Source - Relay	Relay - Destination
8 meter	4 meter	4 meter
6 meter	3 meter	3 meter
4 meter	2 meter	2 meter

Untuk *layout* pengukuran pada ruang AJ403 dapat dilihat pada gambar 3.14.



Gambar 3.14 Layout Ruangan AJ403

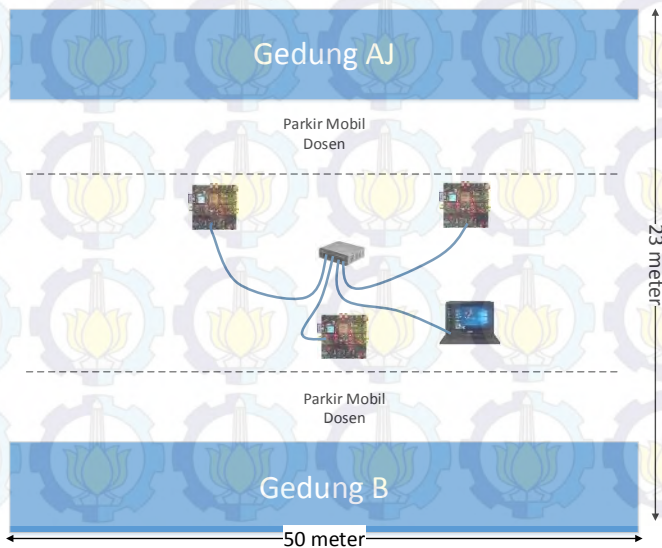
3.5.2 Pengukuran Outdoor

Pada pengukuran *outdoor*, lokasi yang digunakan adalah lapangan parkir dosen Jurusan Teknik Elektro FTI – ITS. Sama seperti *indoor* pengukuran pada *outdoor* juga dilakukan dengan jarak yang berbeda. Variasi jarak yang ditentukan pada pengukuran *outdoor* dapat dilihat pada tabel 3.3.

Tabel 3.3 Variasi Jarak Pengukuran *Outdoor* (Lapangan Parkir Dosen)

<i>Source - Destination</i>	<i>Source - Relay</i>	<i>Relay - Destination</i>
30 meter	15 meter	15 meter
24 meter	12 meter	12 meter
18 meter	9 meter	9 meter
12 meter	6 meter	6 meter
8 meter	4 meter	4 meter
6 meter	3 meter	3 meter
4 meter	2 meter	2 meter

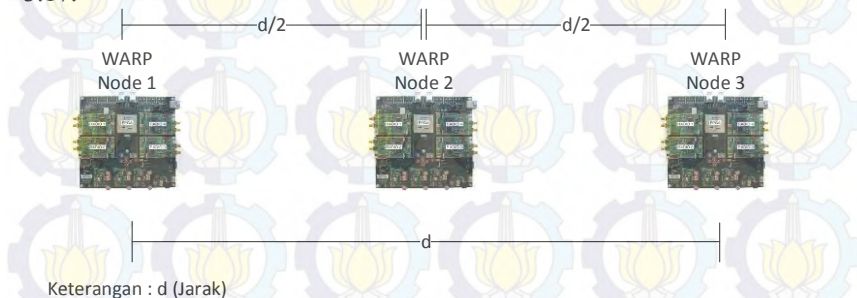
Untuk *layout* pengukuran *outdoor* pada parkir dosen dapat dilihat pada gambar 3.15.



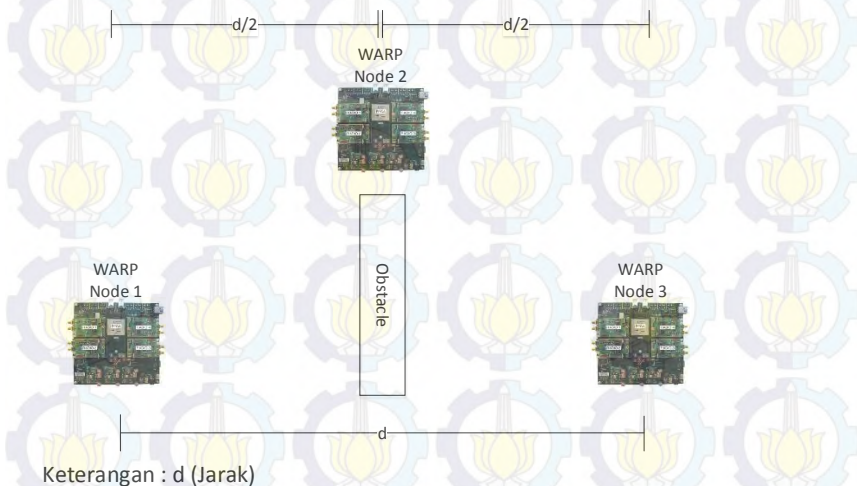
Gambar 3.15 *Layout* Pengukuran di Parkiran Dosen

3.5.3 Kondisi Pengukuran

Pada tugas akhir ini pengukuran dilakukan dengan dua kondisi, *Line of Sight* atau tanpa penghalang, dan *Non Line of Sight* atau dengan penghalang, scenario pengukurannya dapat dilihat pada gambar 3.16 dan 3.17.



Gambar 3.16 Sketsa Pengukuran *Line of Sight*



Gambar 3.17 Sketsa Pengukuran dengan Penghalang

3.6 Perhitungan *Bit Error Rate* (BER)

Bit Error Rate (BER) adalah salah satu parameter yang diukur dalam tugas akhir ini. Perhitungan BER pada tugas akhir ini, data yang masih berupa simbol diubah menjadi deretan bit yang terdiri dari satu baris, kemudian bit tersebut akan dibandingkan oleh bit yang ada disisi

pengirim, dari perbandingan itulah akan dihasilkan nilai BER. Berikut adalah persamaan untuk menghitung BER.

$$\text{Bit Error Rate} = \frac{\text{Bit Salah}}{\text{Bit Dikirim}} \quad (3.1)$$

3.7 Perhitungan *Throughput*

Throughput adalah tingkat kesuksesan dari informasi yang dikirim dalam suatu kanal komunikasi. Pada tugas akhir ini salah satu parameter yang diukur adalah *throughput*, persamaan untuk menghitung *throughput* adalah sebagai berikut [17].

$$\text{Throughput} = \left(\frac{\text{Jumlah Bit yang Benar}}{\text{Waktu Satu Kali Transmisi}} \right) \quad (3.2)$$



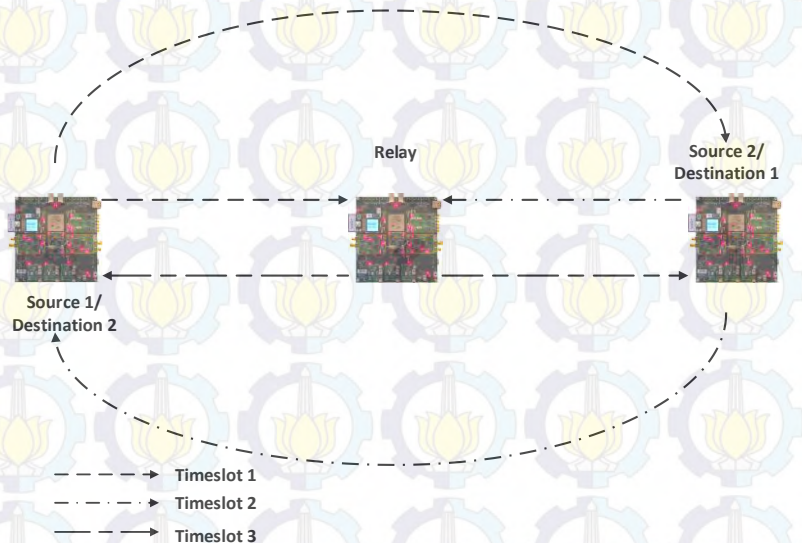
BAB 4

HASIL DAN ANALISA

Pada bagian ini akan dijelaskan hasil implementasi serta analisa dari sistem komunikasi kooperatif *decoding and forward* dengan *network coding* menggunakan *Wireless Open-Access Research Platform*. Analisis hasil implementasi akan merujuk kepada hasil simulasi yang telah dijelaskan pada bab sebelumnya

4.1 Skema Implementasi

Pada implementasi akan digunakan tiga buah *node* WARP, dengan skema seperti gambar 4.1.



Gambar 4.1 Skema Implementasi *Network Coding* pada Sistem Komunikasi Kooperatif DF

Dari gambar diatas, akan diimplementasikan skema *network coding* dengan menggunakan *Wireless Open-Access Research Platform* dengan masing-masing *node* berfungsi sebagai *source, relay*, dan *destination*.

4.2 Analisa Kinerja *Network Coding* pada Sistem Komunikasi Kooperatif (*Indoor*)

Pada bagian ini akan dibahas hasil implementasi *indoor* sistem komunikasi kooperatif dengan *network coding*. Parameter yang dibandingkan pada bagian ini adalah BER dengan pembandingnya adalah sistem komunikasi kooperatif konvensional dan komunikasi jalur langsung. Perbandingan dengan komunikasi jalur langsung dimaksudkan untuk mengetahui kelebihan dari komunikasi *relay*, karena *network coding* adalah komunikasi yg memanfaatkan *relay*. Selain itu kan dibandingkan juga performa sistem komunikasi kooperatif *network coding* dengan yang memakai *coding* dan tidak memakai *coding*. Implementasi sistem komunikasi ini akan dibuat dalam dua kondisi yaitu dengan penghalang (*Non Line of Sight*) dan tanpa penghalang (*Line of Sight*). Untuk pengujian *indoor* dilakukan di ruang AJ403, sementara untuk *outdoor* dilakukan di area parkir dosen Teknik Elektro ITS.

4.2.1 Line of Sight

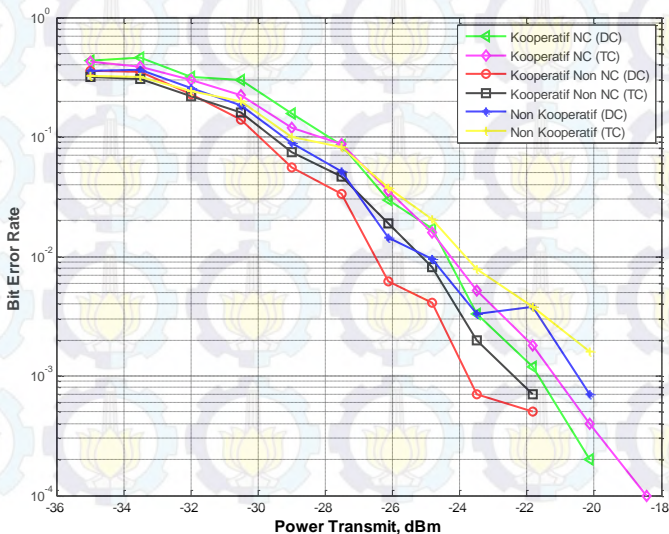
Untuk hasil implementasi dengan kondisi *line of sight* dapat dilihat pada tabel 4.1.

Tabel 4.1 Tabel Perbandingan BER Pada Jarak 8 Meter (*Indoor*)

Gain	Ptx (dBm)	Bit Error Rate					
		Dengan Coding			Tanpa Coding		
		Non Koop	Non NC	Dengan NC	Non Koop	Non NC	Dengan NC
0	-36.1	0.3601	0.3572	0.4376	0.3223	0.3154	0.4296
3	-34.5	0.3621	0.3523	0.4605	0.3166	0.3056	0.3892
6	-32.9	0.2579	0.2269	0.3172	0.2411	0.22	0.3022
9	-31.3	0.185	0.1382	0.3014	0.1991	0.1608	0.2233
12	-29.7	0.0895	0.0557	0.1579	0.0996	0.0742	0.1189
15	-28.2	0.0516	0.0331	0.086	0.0826	0.0464	0.0864
18	-26.6	0.0144	0.0062	0.0299	0.0372	0.0188	0.0351
21	-25.0	0.0095	0.0041	0.017	0.0206	0.0081	0.0159
24	-23.4	0.0033	0.0007	0.0033	0.0079	0.002	0.0052
27	-21.9	0.0038	0.0005	0.0012	0.0038	0.0007	0.0018
30	-20.3	0.0007	0	0.0002	0.0016	0	0.0004
33	-18.7	0	0	0	0	0	0.0001
36	-17.1	0	0	0	0	0	0

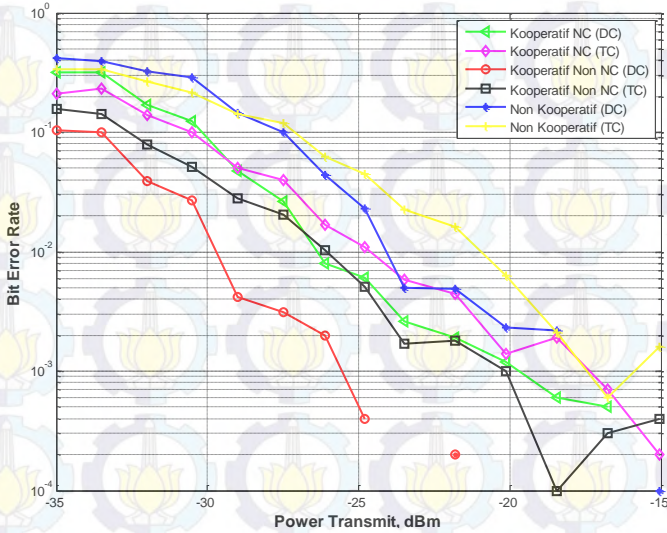
Dari tabel 4.1 diatas dapat dilihat bahwa saat diimplementasikan pada modul WARP. Kinerja komunikasi yang menggunakan dengan *relay* mempunyai kinerja yang lebih baik dibandingkan dengan komunikasi jalur langsung. Hal ini dapat terlihat dari data yang dipaparkan diatas, misal pada saat daya pancar mencapai nilai -20.3 dBm, BER pada sistem kooperatif konvensional sudah mencapai nilai 0, sedangkan pada sistem komunikasi SISO BER masih ada di nilai 0.0007 sementara sistem komunikasi kooperatif *network coding* bernilai 0.0002. Sementara pada saat daya pancar mencapai level -18.7 dBm nilai BER sudah mencapai 0 semua. Namun jika dilihat dari kurva keseluruhan kinerja komunikasi non kooperatif mempunyai kinerja yang kurang baik dibandingkan sistem komunikasi yang menggunakan *relay*.

Sebagai pembandingan, sistem komunikasi ini juga akan dibandingkan dengan sistem komunikasi tanpa *coding*. Dapat dilihat bahwa komunikasi non kooperatif juga mempunyai performa yg kurang baik dibanding *network coding* dan kooperatif konvensional. Hal ini dapat dilihat, pada saat nilai daya pancar mencapai -20.3 dBm. BER untuk komunikasi SISO berada di nilai 0.0016, sedangkan pada komunikasi kooperatif BER mencapai nilai 0, dan pada komunikasi kooperatif *network coding* BER berada pada nilai 0.0004. Untuk kurva hasil pengukuran pada jarak 8 meter dapat dilihat pada gambar 4.2.

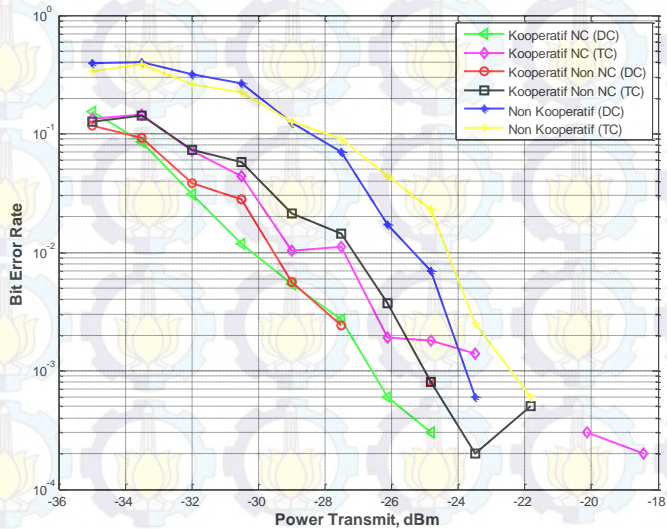


Gambar 4.2 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter

Untuk hasil pengukuran pada jarak 6 dan 4 meter kondisi *line of sight* atau tanpa penghalang, dapat dilihat pada gambar 4.3 dan 4.4.



Gambar 4.3 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter



Gambar 4.4 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter

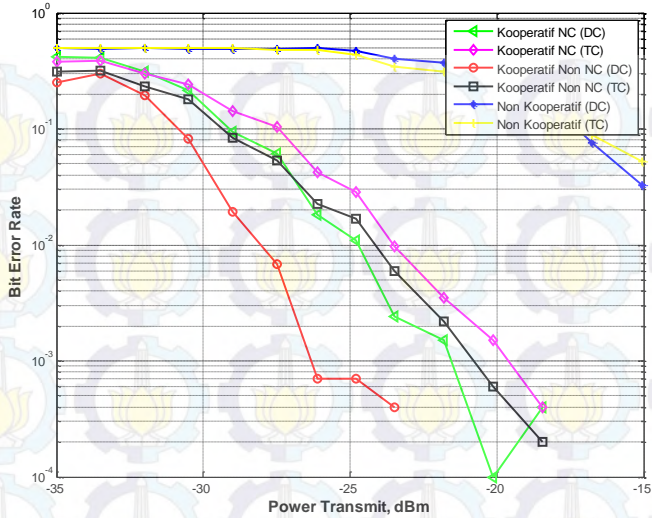
4.2.2 Non Line of Sight

Untuk hasil implementasi dengan kondisi *non line of sight* dapat dilihat pada tabel 4.2.

Tabel 4.2 Tabel Perbandingan BER Pada Jarak 8 Meter (*Indoor*)

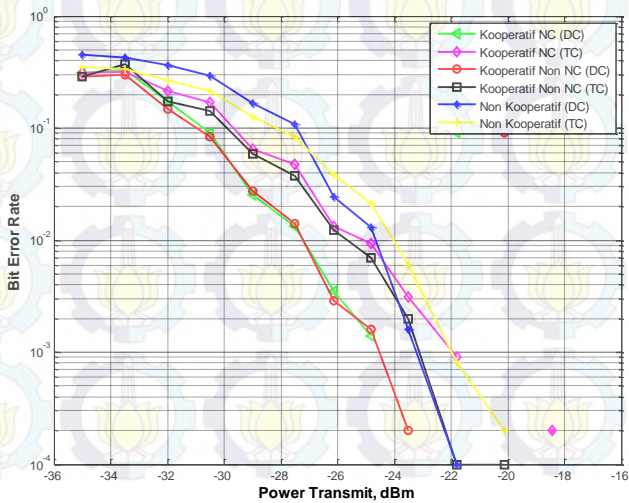
Gain	Ptx (dBm)	Bit Error Rate					
		Dengan Coding			Tanpa Coding		
		Non Koop	Non NC	Dengan NC	Non Koop	Non NC	Dengan NC
0	-36.1	0.4958	0.2531	0.4209	0.4997	0.3101	0.3824
3	-34.5	0.4912	0.2974	0.4103	0.5026	0.3147	0.3878
6	-32.9	0.4971	0.1937	0.3132	0.4981	0.2313	0.2974
9	-31.3	0.4926	0.0825	0.2134	0.4965	0.1804	0.2427
12	-29.7	0.4904	0.0193	0.0947	0.5011	0.0845	0.1411
15	-28.2	0.4919	0.0068	0.0616	0.4798	0.0539	0.1036
18	-26.6	0.4971	0.0007	0.0182	0.4771	0.0224	0.0421
21	-25.0	0.4673	0.0007	0.011	0.4335	0.0169	0.0284
24	-23.4	0.4044	0.0004	0.0024	0.3405	0.006	0.0097
27	-21.9	0.3737	0	0.0015	0.3109	0.0022	0.0035
30	-20.3	0.2515	0	0.0001	0.2221	0.0006	0.0015
33	-18.7	0.1958	0	0.0004	0.1642	0.0002	0.0004
36	-17.1	0.0759	0	0	0.0895	0	0

Dari tabel 4.2, dapat dilihat bahwa untuk kondisi *non line of sight* atau dengan penghalang, kinerja dari komunikasi relay masih lebih baik dibandingkan komunikasi jalur langsung. Hal ini dapat terlihat dari data diatas yang sudah disajikan dalam bentuk grafik, dapat terlihat bahwa komunikasi non kooperatif nilai BER-nya tidak mencapai 0, lain halnya dengan komunikasi *network coding* dan kooperatif konvensional yang sudah mencapai 0 pada saat daya pancar mencapai -17.1 dBm. Begitu pula dengan komunikasi yang tanpa menggunakan *coding*, performa komunikasi *relay* lebih baik dibandingkan dengan komunikasi jalur langsung. Hal ini dapat terlihat saat nilai daya pancar mencapai -17.1 dBm, nilai BER pada komunikasi non kooperatif masih berada di kisaran 0.0895 dan nilai BER untuk komunikasi kooperatif dan *network coding* sudah mencapai 0. Untuk kurva hasil implementasi pada jarak 8 meter dengan kondisi *non line of sight* atau dengan penghalang dapat dilihat pada gambar 4.5.

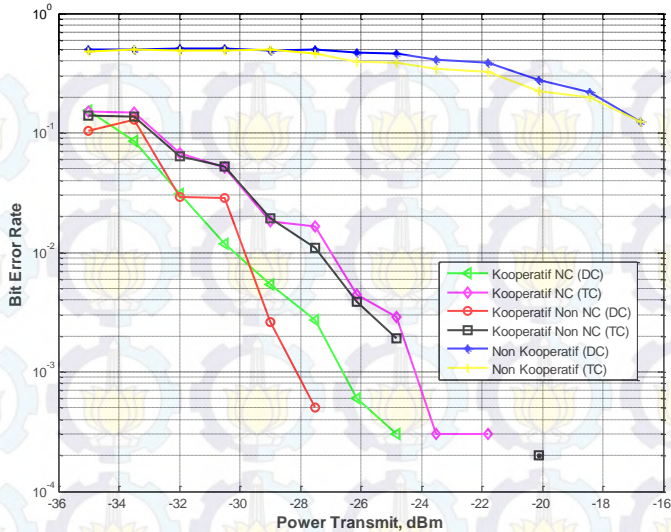


Gambar 4.5 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter

Untuk hasil pengukuran pada jarak 6 dan 4 meter kondisi *non line of sight* atau dengan penghalang, dapat dilihat pada gambar 4.6 dan 4.7.



Gambar 4.6 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter



Gambar 4.7 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter

4.3 Analisa Kinerja *Network Coding* pada Sistem Komunikasi Kooperatif (*Outdoor*)

Pada bagian ini akan dibahas mengenai hasil implementasi pada kondisi *outdoor*. Seperti yang telah dijelaskan pada bagian hasil implementasi pada kondisi *indoor*. Pada bagian ini kinerja sistem komunikasi akan diuji dengan BER sebagai parameternya. Pada bagian ini akan dibandingkan juga sistem komunikasi kooperatif *network coding*, kooperatif tanpa *network coding* dan jalur langsung (*Point to Point*). Pada pengujian *outdoor* akan dilihat pula performanya dengan mengatur jarak terjauh yang dapat dijangkau, yaitu 30 m.

4.3.1 Line of Sight

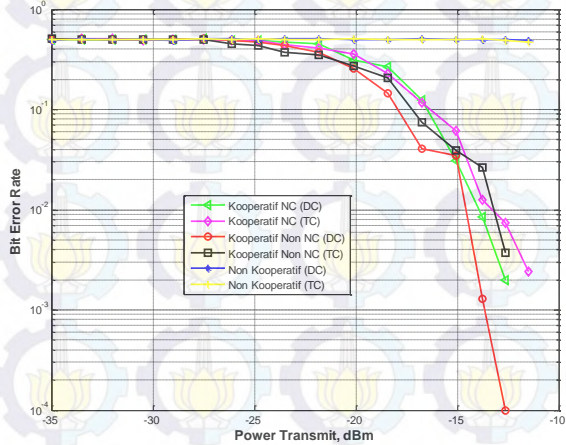
Untuk hasil implementasi sistem komunikasi dengan kondisi *line of sight* dapat dilihat pada tabel 4.3.

Tabel 4.3 Tabel Perbandingan BER Pada Jarak 30 Meter (*Outdoor*)

Gain	Ptx (dBm)	Bit Error Rate					
		Dengan Coding			Tanpa Coding		
		Non Koop	Non NC	Dengan NC	Non Koop	Non NC	Dengan NC
0	-36.1	0.4910	0.4971	0.5014	0.497	0.5058	0.5015
3	-34.5	0.5061	0.5058	0.4984	0.4989	0.4959	0.5084
6	-32.9	0.4972	0.5011	0.5011	0.4981	0.4991	0.5081
9	-31.3	0.4999	0.5028	0.4997	0.4959	0.4962	0.4909
12	-29.7	0.4898	0.4986	0.4975	0.5008	0.495	0.4942
15	-28.2	0.5047	0.5045	0.5069	0.5038	0.4996	0.4941
18	-26.6	0.5048	0.4847	0.491	0.5029	0.4518	0.491
21	-25.0	0.4980	0.4658	0.4864	0.5073	0.4315	0.4852
24	-23.4	0.5062	0.4375	0.4746	0.4992	0.3696	0.4394
27	-21.9	0.5078	0.3715	0.4527	0.4971	0.3502	0.4063
30	-20.3	0.5029	0.2559	0.312	0.5041	0.2733	0.3588
33	-18.7	0.4954	0.1446	0.2645	0.4996	0.2064	0.2273
36	-17.1	0.5046	0.0407	0.123	0.4978	0.0749	0.118
39	-15.1	0.4984	0.0348	0.0311	0.4992	0.0391	0.0617
42	-14.0	0.4999	0.0013	0.0084	0.5041	0.0264	0.0125
45	-12.4	0.5028	0.0001	0.002	0.49	0.0037	0.0074
48	-10.8	0.4931	0.	0	0.4665	0	0.0024

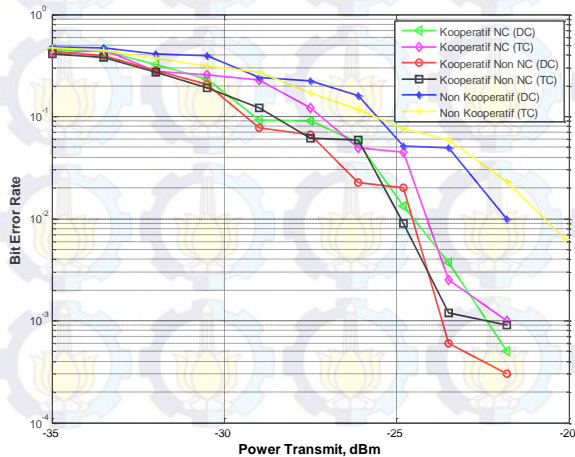
Pada pengujian *outdoor* yang dilakukan diluar ruangan, sistem komunikasi kooperatif akan diuji coba dengan jarak terjauh yang dapat dijangkau, dari pengujian yang telah dilakukan jarak terjauh yang dapat dijangkau adalah 30 meter dengan *relay* berada diantara *source* dan *destination*. Untuk data *outdoor* dengan kondisi *line of sight* dapat dilihat pada tabel 4.3 dari data tersebut dapat dilihat bahwa komunikasi *relay* tetap mempunyai kinerja yang lebih baik dibandingkan dengan komunikasi jalur langsung. Misalkan pada saat nilai daya pancar sudah mencapai -10.8 dBm BER pada sistem komunikasi *relay* yaitu sistem komunikasi kooperatif dan *network coding* sudah mencaoai nilai 0, sedangkan komunikasi jalur langsung bernilai 0.4931, begitupun dengan komunikasi yang tanpa menggunakan coding. Pada saat daya pancar mencapai -10.8 BER untuk sistem komunikasi kooperatif konvensional sudah mencapai 0, sementara untuk komunikasi kooperatif *network coding* mempunyai nilai 0.0024 dan untuk komunikasi jalur langsung

masih berada diangka 0.4665. Pengujian dengan jarak terjauh ini dimaksudkan untuk menguji kualitas kinerja sistem komunikasi kooperatif *network coding* berdasarkan jarak. Untuk kurva hasil implementasi pada jarak 30 meter dengan kondisi *line of sight* atau tanpa penghalang dapat dilihat pada gambar 4.8.

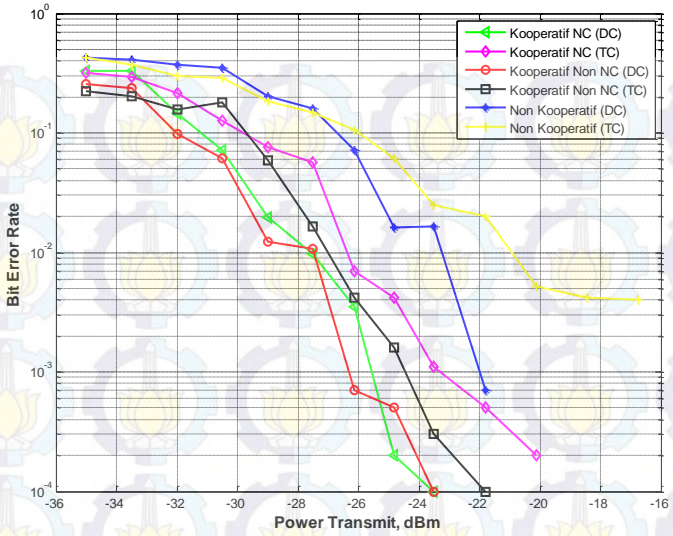


Gambar 4.8 Grafik Kurva Hasil Implementasi BER Jarak 30 Meter

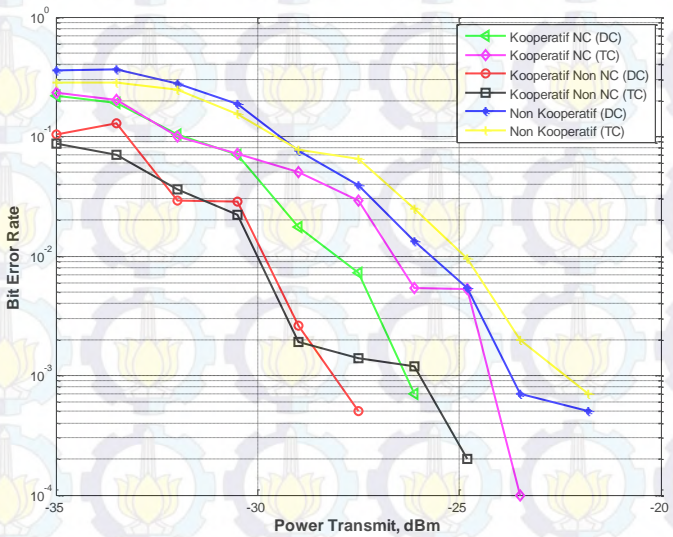
Untuk pengukuran *outdoor* pada jarak 8,6, dan 4 dapat dilihat pada gambar 4.9 – 4.11.



Gambar 4.9 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter



Gambar 4.10 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter



Gambar 4.11 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter

4.3.2 Non Line of Sight

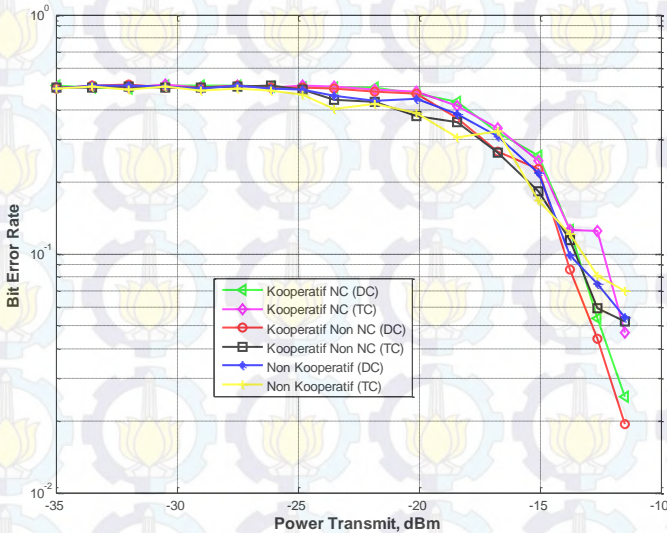
Untuk hasil implementasi sistem komunikasi dengan kondisi *non line of sight* dapat dilihat pada tabel 4.4.

Tabel 4.4 Tabel Perbandingan BER Pada Jarak 30 Meter (*Outdoor*)

Gain	Ptx (dBm)	Bit Error Rate					
		Dengan Coding			Tanpa Coding		
		Non Koop	Non NC	Dengan NC	Non Koop	Non NC	Dengan NC
0	-36.1	0.4931	0.4923	0.5069	0.4931	0.4962	0.4979
3	-34.5	0.5038	0.5045	0.4942	0.5	0.4948	0.5022
6	-32.9	0.5038	0.5088	0.4916	0.4887	0.5021	0.4974
9	-31.3	0.5025	0.4978	0.5054	0.5	0.4981	0.509
12	-29.7	0.4936	0.4956	0.508	0.4832	0.4967	0.4975
15	-28.2	0.5061	0.5069	0.5078	0.4918	0.5018	0.5067
18	-26.6	0.4938	0.4966	0.5002	0.4801	0.5055	0.498
21	-25.0	0.4845	0.4956	0.497	0.4641	0.4855	0.5044
24	-23.4	0.4583	0.4906	0.4956	0.4042	0.4403	0.5003
27	-21.9	0.4368	0.476	0.4979	0.4252	0.4312	0.4857
30	-20.3	0.4472	0.4663	0.4699	0.3886	0.3788	0.4778
33	-18.7	0.3833	0.3703	0.4332	0.3076	0.3572	0.4157
36	-17.1	0.3095	0.2674	0.3224	0.3251	0.2653	0.3348
39	-15.1	0.2174	0.2262	0.2586	0.1675	0.1834	0.2447
42	-14.0	0.0987	0.0856	0.1253	0.1212	0.1144	0.1256
45	-12.4	0.0749	0.0443	0.0536	0.0812	0.0591	0.1241
48	-10.8	0.0543	0.0194	0.0252	0.0696	0.0523	0.0469

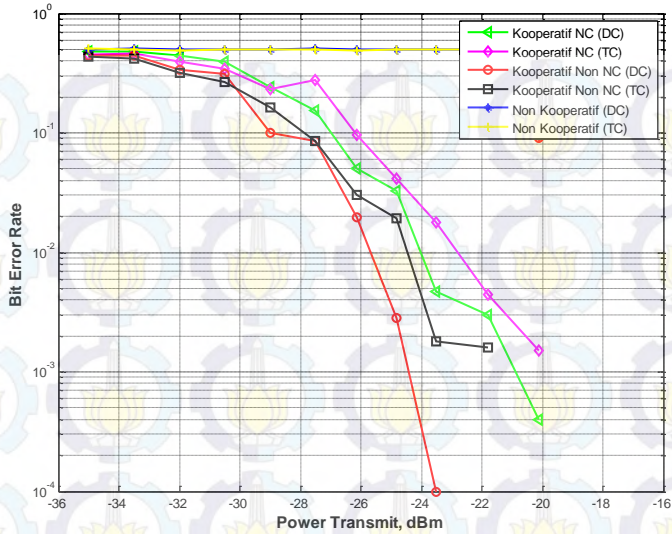
Pada pengujian *outdoor* dengan kondisi diberi penghalang atau *non line of sight*. Sistem komunikasi kooperatif dengan *network coding* akan diujikan dengan jarak terjauh juga seperti pada pengujian dengan kondisi LOS. Hasil dari pengujian ini dapat dilihat pada tabel 4.4. Dari data tabel tersebut dapat dilihat bahwa kinerja komunikasi yang memanfaatkan *relay* masih lebih baik dibandingkan dengan komunikasi jalur langsung. Misalkan pada komunikasi kooperatif *network coding* saat daya pancar mencapai nilai -10.8 dBm, nilai BER yang didapat adalah 0.0252, sedangkan pada komunikasi kooperatif adalah 0.0194 dan pada komunikasi jalur langsung nilai BER yang didapat adalah 0.0543. Hal yang sama juga terjadi pada komunikasi kooperatif *network coding* tanpa

menggunakan *coding* dari data tersebut dapat terlihat pada komunikasi *relay* juga mempunyai kinerja yang lebih baik dibandingkan dengan komunikasi jalur langsung. Misal pada komunikasi kooperatif *network coding* saat mencapai daya pancar -10.8 dBm nilai BER berada pada 0.0469, sedangkan pada kooperatif konvensional 0.0523 dan pada jalur langsung 0.0649. Untuk kurva hasil implementasi pada jarak 30 meter dengan kondisi *non line of sight* atau dengan penghalang dapat dilihat pada gambar 4.12.

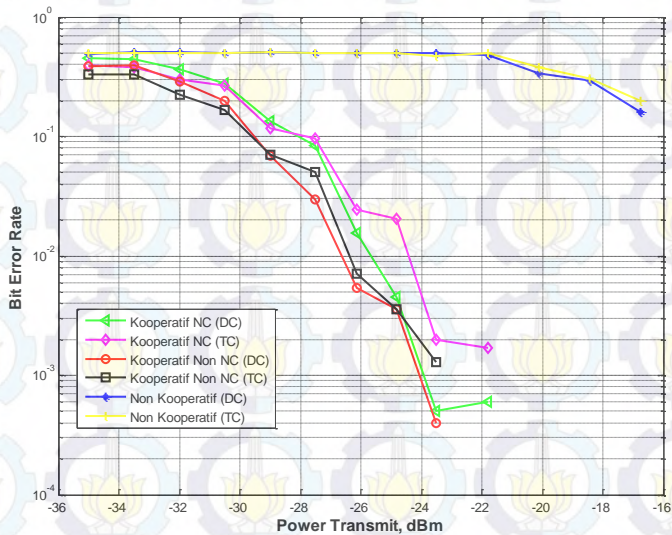


Gambar 4.12 Grafik Kurva Hasil Implementasi BER Jarak 30 Meter

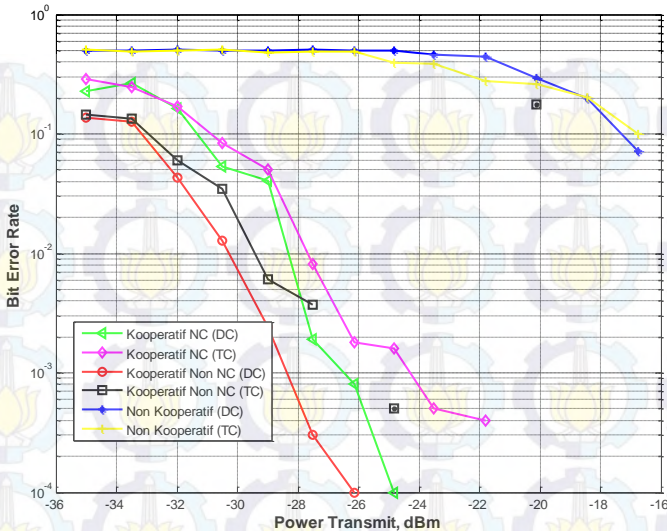
Untuk pengukuran pada jarak 4,6, dan 8 meter dapat dilihat pada gambar 4.13 – 4.15.



Gambar 4.13 Grafik Kurva Hasil Implementasi BER Jarak 8 Meter



Gambar 4.14 Grafik Kurva Hasil Implementasi BER Jarak 6 Meter



Gambar 4.15 Grafik Kurva Hasil Implementasi BER Jarak 4 Meter

4.4 Analisa Kinerja *Network Coding* pada Sistem Komunikasi Kooperatif Berdasarkan Jarak

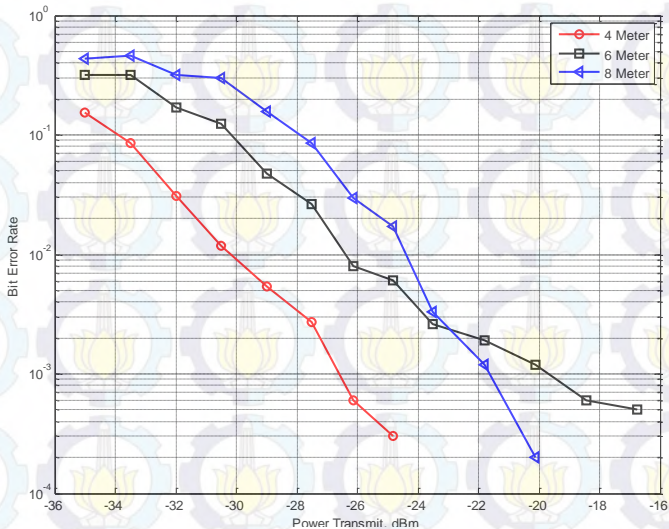
Pada bab 3 telah dibahas bahwa tugas akhir ini akan diimplementasikan dengan variasi jarak. Pada bagian ini akan dibahas kinerja sistem komunikasi kooperatif berdasarkan variasi jarak yang sudah ditentukan pada bagian sebelumnya. Berikut adalah hasil kinerja sistem komunikasi kooperatif *network coding* berdasarkan variasi jarak.

4.4.1 Perbandingan Kinerja pada Kondisi *Indoor*

Pada bagian ini akan dibahas mengenai kinerja sistem komunikasi kooperatif berdasarkan jarak yang telah ditentukan pada kondisi *indoor*. Pada kondisi *indoor* akan ada dua kondisi lagi, yaitu dengan penghalang dan tanpa penghalang. Hasil implementasi dengan perbandingan variasi jarak *indoor* pada kondisi *line of sight* atau tanpa penghalang dapat dilihat pada tabel 4.5.

Tabel 4.5 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif *Network Coding* dengan Variasi Jarak pada Kondisi LOS (*Indoor*)

Gain	P _{tx} (dBm)	Bit Error Rate		
		4 Meter	6 Meter	8 Meter
0	-36.1	0.1538	0.3194	0.4376
3	-34.5	0.0858	0.3165	0.4605
6	-32.9	0.0306	0.1682	0.3172
9	-31.3	0.0118	0.1243	0.3014
12	-29.7	0.0054	0.0475	0.1579
15	-28.2	0.0027	0.0266	0.0860
18	-26.6	0.0006	0.0080	0.0299
21	-25.0	0.0003	0.0061	0.0170
24	-23.4	0	0.0026	0.0033
27	-21.9	0	0.0019	0.0012
30	-20.3	0	0.0012	0.0002
33	-18.7	0	0.0006	0
36	-17.1	0	0.0005	0



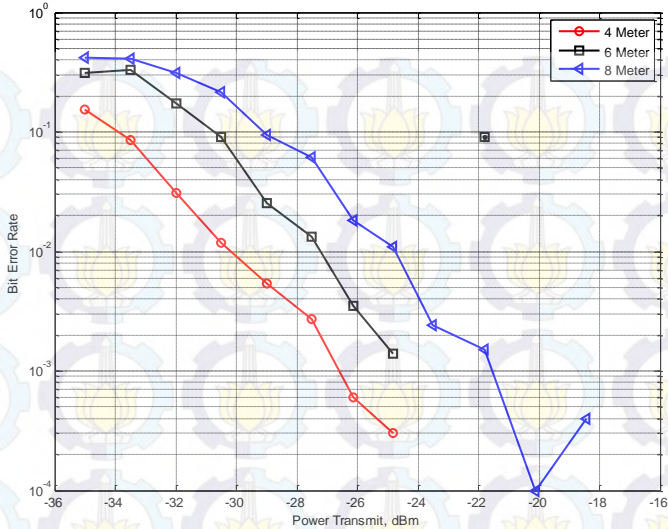
Gambar 4.16 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi LOS (*Indoor*)

Dari data pada tabel 4.5 dan gambar 4.16 diatas, dapat terlihat bahwa semakin dekat jaraknya maka sistem komunikasi kooperatif *network coding* semakin bagus. Misalkan pada saat daya pancar -25.0 dBm, BER pada jarak 4 meter bernilai 0.0003 sedangkan untuk 6 meter bernilai 0.0061, sementara untuk 8 meter 0.0170.

Untuk hasil analisa perbandingan jarak komunikasi kooperatif *network coding* pada *indoor* dengan kondisi diberi penghalang dapat dilihat pada tabel 4.6 dan gambar 4.17.

Tabel 4.6 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif *Network Coding* dengan Variasi Jarak pada Kondisi NLOS (*Indoor*)

Gain	Ptx (dBm)	Bit Error Rate		
		4 Meter	6 Meter	8 Meter
0	-36.1	0.1538	0.3127	0.4209
3	-34.5	0.0858	0.3290	0.4103
6	-32.9	0.0306	0.1747	0.3132
9	-31.3	0.0118	0.0910	0.2134
12	-29.7	0.0054	0.0255	0.0947
15	-28.2	0.0027	0.0134	0.0616
18	-26.6	0.0006	0.0035	0.0182
21	-25.0	0.0003	0.0014	0.0110
24	-23.4	0	0	0.0024
27	-21.9	0	0.0898	0.0015
30	-20.3	0	0	0.0001
33	-18.7	0	0	0.0004
36	-17.1	0	0	0



Gambar 4.17 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi NLOS (*Indoor*)

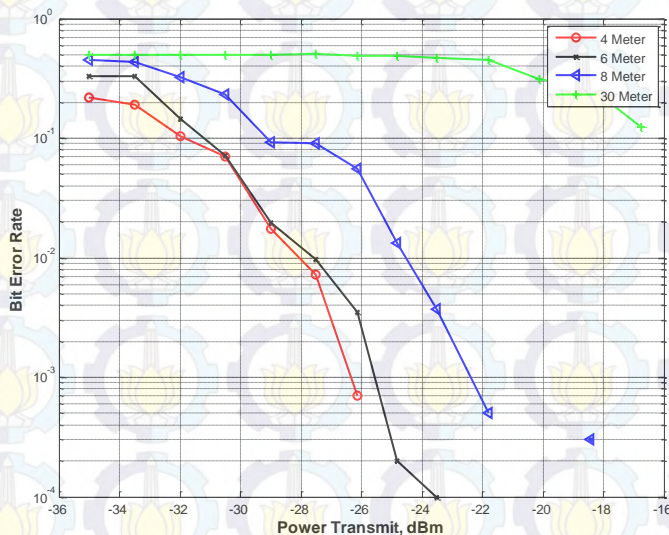
Dari data diatas, kinerja sistem komunikasi kooperatif *network coding* lebih baik jika jaraknya didekatkan juga meskipun dalam kondisi NLOS. Terlihat dari grafik diatas, misalkan pada saat daya pancar -25.0 dBm, untuk jarak 4 meter nilai BER adalah 0.0003, sedangkan untuk jarak 6 meter adalah 0.0014 dan untuk jarak 8 meter adalah 0.0110.

4.4.2 Perbandingan Kinerja pada Kondisi *Outdoor*

Pada bagian ini akan dibahas mengenai hasil kinerja berdasarkan variasi jarak pada sistem komunikasi kooperatif *network coding* dengan kondisi *outdoor*. Sama seperti *indoor*, pada *outdoor* juga akan dikondisikan dengan penghalang dan tanpa penghalang. Hasil impelementasi dengan perbandingan variasi jarak *indoor* pada kondisi *non line of sight* atau tanpa penghalang dapat dilihat pada tabel 4.7.

Tabel 4.7 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif *Network Coding* dengan Variasi Jarak pada Kondisi LOS (*Outdoor*)

Gain	Ptx (dBm)	Bit Error Rate			
		4 Meter	6 Meter	8 Meter	30 Meter
0	-36.1	0.2209	0.3307	0.4508	0.4508
3	-34.5	0.1920	0.3311	0.4324	0.4324
6	-32.9	0.1048	0.1463	0.3271	0.3271
9	-31.3	0.0699	0.0716	0.2307	0.2307
12	-29.7	0.0176	0.0197	0.0921	0.0921
15	-28.2	0.0073	0.0098	0.0963	0.0903
18	-26.6	0.0007	0.0035	0.0552	0.0552
21	-25.0	0	0.0002	0.0134	0.0134
24	-23.4	0	0.0001	0.0037	0.0037
27	-21.9	0	0	0.0005	0.0005
30	-20.3	0	0	0.0002	0
33	-18.7	0	0	0.0003	0.0003
36	-17.1	0	0	0	0

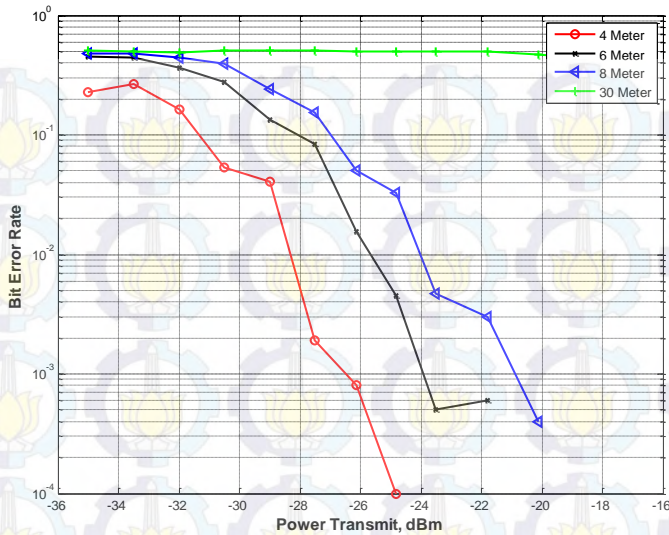


Gambar 4.18 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi LOS (*Outdoor*)

Berdasarkan tabel 4.7 dan gambar 4.18 kinerja sistem komunikasi kooperatif pada *outdoor* dengan kondisi LOS juga dipengaruhi oleh jarak hal tersebut terlihat dari hasil perbandingan pengukuran dengan variasi jarak pada lingkungan *outdoor*, dengan jarak terjauh 30 meter, dan jarak terdekat 4 meter. Misalkan pada saat kondisi daya pancar -25.0 dBm BER sistem komunikasi kooperatif *network coding* pada jarak 4 meter bernilai 0, sedangkan untuk jarak 6 meter nilai BER adalah 0.0002 untuk jarak 8 meter 0.0134 sementara untuk jarak terjauh yang diujikan nilai BER adalah 0.4864. Untuk perbandingan nilai BER pada kondisi NLOS dapat dilihat pada gambar dan tabel berikut.

Tabel 4.8 Tabel Perbandingan Kinerja Sistem Komunikasi Kooperatif *Network Coding* dengan Variasi Jarak pada Kondisi NLOS (*Outdoor*)

Gain	Ptx (dBm)	Bit Error Rate			
		4 Meter	6 Meter	8 Meter	30 Meter
0	-36.1	0.2257	0.4528	0.4818	0.5069
3	-34.5	0.2668	0.4461	0.4755	0.4942
6	-32.9	0.1623	0.3664	0.4416	0.4916
9	-31.3	0.0529	0.2757	0.3980	0.5054
12	-29.7	0.0405	0.1342	0.2394	0.5080
15	-28.2	0.0019	0.0836	0.1542	0.5078
18	-26.6	0.0008	0.0154	0.0503	0.5002
21	-25.0	0.0001	0.0045	0.0327	0.4970
24	-23.4	0	0.0005	0.0047	0.4956
27	-21.9	0	0.0006	0.0030	0.4979
30	-20.3	0	0	0.0004	0.4699
33	-18.7	0	0	0	0.4332
36	-17.1	0	0	0	0.3224



Gambar 4.19 Grafik Perbandingan BER Berdasarkan Variasi Jarak Pada Kondisi NLOS (*Outdoor*)

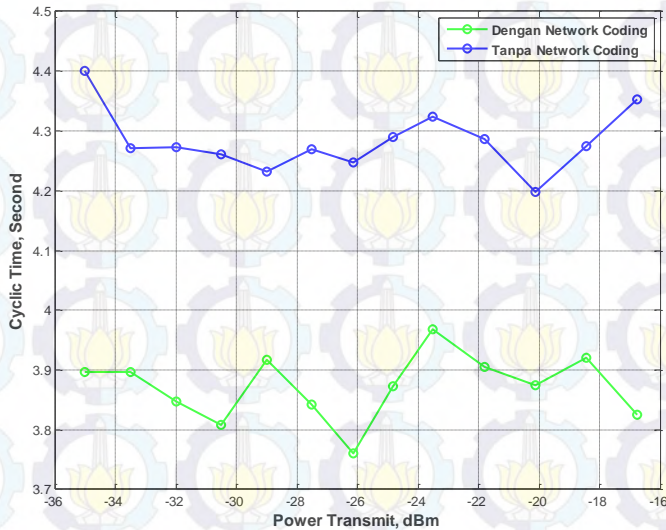
Berdasarkan tabel 4.8 dan gambar 4.19 ditunjukkan hasil yang sama, bahwa sistem komunikasi kooperatif *network coding* akan lebih baik apabila jaraknya didekatkan. Misalkan pada saat daya pancar berada pada level -23.4 dBm nilai BER untuk jarak 4 meter sudah mencapai 0, sedangkan untuk jarak 6 meter adalah 0.0045, kemudian pada jarak 8 meter nilai BER adalah 0.0327, dan untuk jarak terjauhnya adalah 0.4970.

4.5 Perbandingan *Throughput* Antara Sistem Komunikasi Kooperatif Dengan dan Tanpa *Network Coding*

Salah satu kelebihan komunikasi *network coding* adalah peningkatan *throughput* dibandingkan dengan komunikasi konvensional. Pada bagian ini akan dibahas hasil implementasi *throughput* dari sistem komunikasi kooperatif dengan skema *network coding*. Hasil perhitungan *throughput* mengacu pada hasil BER yang didapat seperti pada persamaan (3.2) dan berdasarkan waktu satu kali transmisi. Hasil implementasi dengan parameter *throughput* dapat dilihat pada gambar berikut.

Tabel 4.9 Tabel Perbandingan *Cyclic Time* pada Sistem Komunikasi Kooperatif yang Memakai *Network Coding* dengan yang Tidak

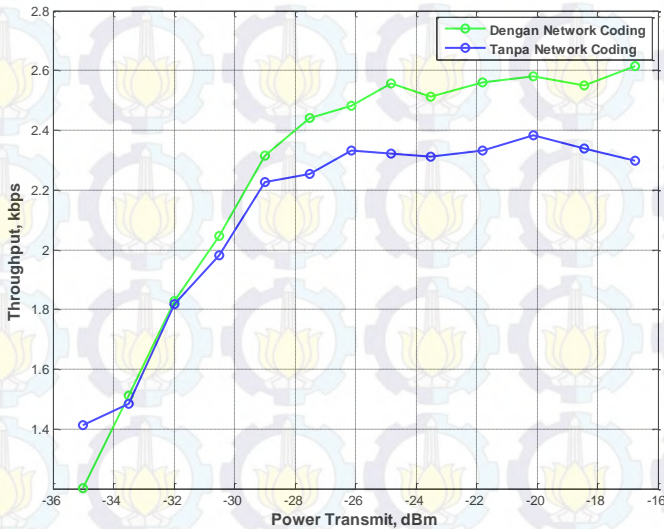
Gain	Ptx (dBm)	<i>Cyclic Time</i> (Second)	
		Dengan <i>Network Coding</i>	Tanpa <i>Network Coding</i>
0	-36.1	3.8972	4.4003
3	-34.5	3.8972	4.2719
6	-32.9	3.8477	4.2735
9	-31.3	3.8081	4.2609
12	-29.7	3.9177	4.2312
15	-28.2	3.842	4.2695
18	-26.6	3.7607	4.2479
21	-25.0	3.8732	4.289
24	-23.4	3.9677	4.3235
27	-21.9	3.9044	4.2859
30	-20.3	3.8743	4.198
33	-18.7	3.9199	4.2749
36	-17.1	3.825	4.3532



Gambar 4.20 Grafik *Cyclic Time* pada Kondisi *Indoor*

Tabel 4.10 Tabel Perbandingan *Throughput* Antara Sistem Komunikasi yang Memakai *Network Coding* dengan yang Tidak (*Indoor*)

Gain	Ptx (dBm)	Throughput (kbps)	
		Dengan Network Coding	Tanpa Network Coding
0	-36.1	1.2003	1.4136
3	-34.5	1.5108	1.4839
6	-32.9	1.8273	1.8184
9	-31.3	2.0467	1.9825
12	-29.7	2.3133	2.2254
15	-28.2	2.4396	2.2541
18	-26.6	2.4828	2.3311
21	-25.0	2.555	2.3218
24	-23.4	2.5113	2.312
27	-21.9	2.561	2.3325
30	-20.3	2.5811	2.3821
33	-18.7	2.5511	2.3392
36	-17.1	2.6144	2.2972

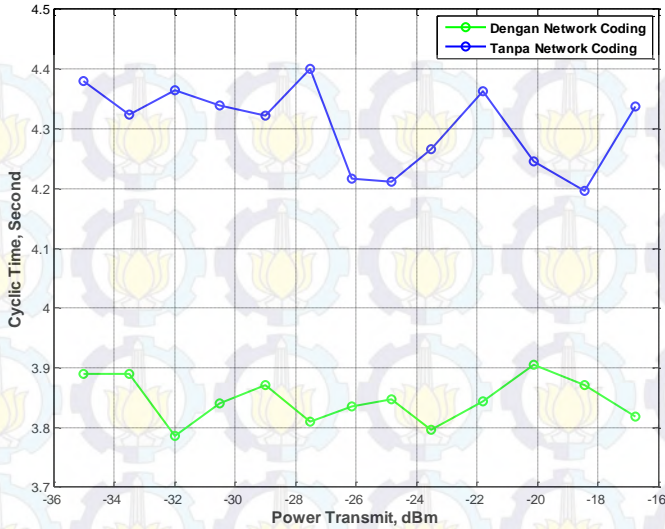


Gambar 4.21 Grafik Kurva *Throughput* pada Kondisi *Indoor*

Dari kurva tersebut diatas, dapat terlihat bahwa *throughput* komunikasi kooperatif dengan *network coding* terlihat lebih baik dibandingkan komunikasi kooperatif konvensional. Misalnya pada saat daya pancar hampir mencapai -17.1 dB komunikasi dengan *network coding* menghasilkan *throughput* 2.6 kbps, sedangkan untuk komunikasi kooperatif konvensional hanya menghasilkan *throughput* 2.2 kbps

Tabel 4.11 Tabel Perbandingan *Cyclic Time* pada Sistem Komunikasi Kooperatif yang Memakai *Network Coding* dengan yang Tidak

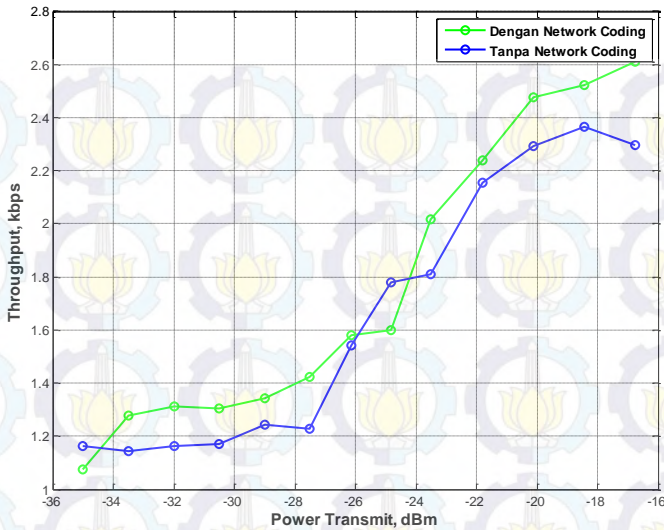
Gain	Ptx (dBm)	<i>Cyclic Time (Second)</i>	
		Dengan Network Coding	Tanpa Network Coding
0	-36.1	3.8897	4.3801
3	-34.5	3.8897	4.3238
6	-32.9	3.7858	4.3647
9	-31.3	3.8409	4.3386
12	-29.7	3.871	4.3217
15	-28.2	3.8104	4.4012
18	-26.6	3.8358	4.2174
21	-25.0	3.8477	4.212
24	-23.4	3.7958	4.2668
27	-21.9	3.8431	4.3638
30	-20.3	3.9052	4.2458
33	-18.7	3.871	4.1963
36	-17.1	3.8183	4.3381



Gambar 4.22 Grafik *Cyclic Time* pada Kondisi *Outdoor*

Tabel 4.12 Tabel Perbandingan *Throughput* Antara Sistem Komunikasi yang Memakai *Network Coding* dengan yang Tidak (*Outdoor*)

Gain	Ptx (dBm)	Throughput (kbps)	
		Dengan Network Coding	Tanpa Network Coding
0	-36.1	1.0764	1.1641
3	-34.5	1.2772	1.1423
6	-32.9	1.3131	1.1625
9	-31.3	1.3031	1.1713
12	-29.7	1.342	1.2446
15	-28.2	1.4237	1.2285
18	-26.6	1.5812	1.5408
21	-25.0	1.5991	1.7801
24	-23.4	2.0156	1.8107
27	-21.9	2.2396	2.1564
30	-20.3	2.4754	2.294
33	-18.7	2.5216	2.364
36	-17.1	2.6093	2.2982



Gambar 4.23 Grafik Kurva *Throughput* pada Kondisi *Outdoor*

Dari gambar diatas, terlihat juga pada saat kondisi *outdoor* kinerja *throughput network coding* terlihat lebih baik. Misalkan pada saat daya pancar hampir mencapai -17.1 dBm, *throughput* yang dicapai komunikasi dengan *network coding* sekitar 2.6 kbps. Sedangkan pada komunikasi konvensional *throughput* yang dihasilkan berada pada nilai 2.4 kbps.



BAB 5 PENUTUP

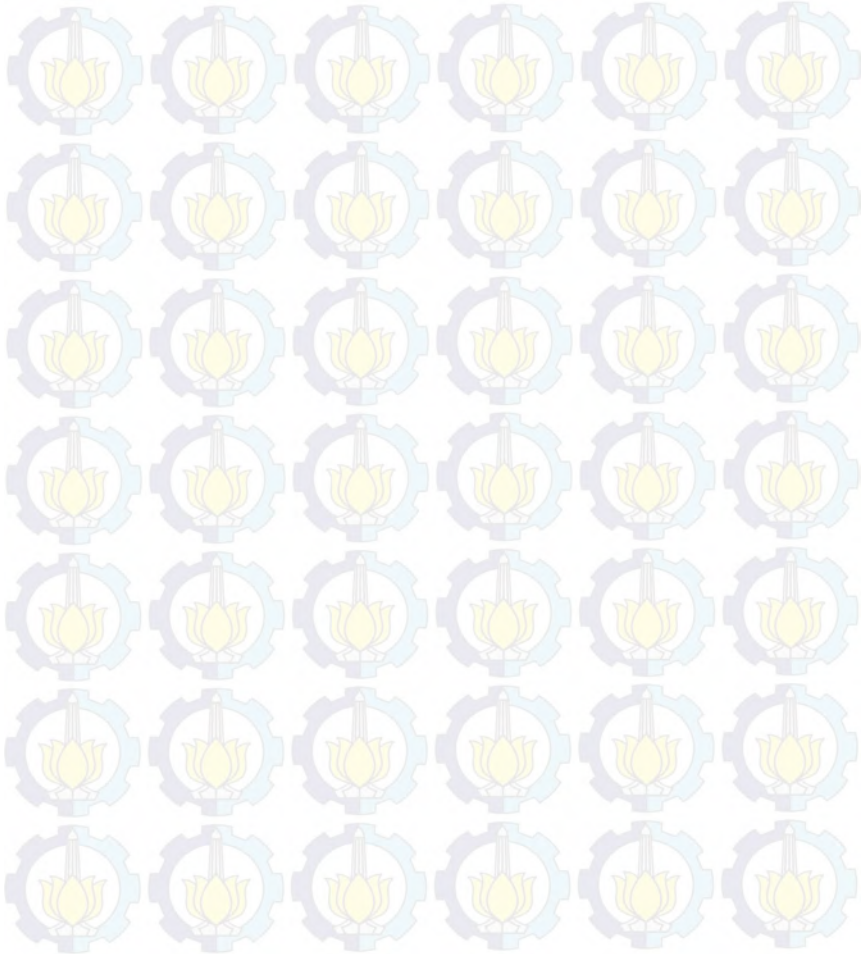
5.1 Kesimpulan

Dari penelitian diatas, dapat ditarik beberapa kesimpulan terkait tugas akhir ini :

1. Sistem komunikasi kooperatif *network coding* yang menggunakan kode konvolusi mempunyai kinerja lebih baik dibandingkan sistem komunikasi yang tidak menggunakan kode konvolusi, seperti pada jarak 8 meter *indoor* saat daya pancar berada pada level -23.4 dBm nilai BER untuk kooperatif *network coding* dengan kode konvolusi adalah 0.0033 sedangkan yang tanpa kode konvolusi adalah 0.0052
2. Kinerja skema *network coding* pada sistem komunikasi kooperatif apabila ditinjau dari *bit error rate* (BER) menghasilkan nilai yang tidak jauh berbeda, seperti pada jarak 8 meter saat daya pancar berada pada level -18.7 dBm berada pada nilai 0.0002, sementara untuk kooperatif tanpa *network coding* berada pada nilai 0
3. Penggunaan skema *network coding* pada komunikasi kooperatif lebih menguntungkan dibandingkan dengan skema kooperatif konvensional dikarenakan dapat menghemat waktu transmisi data lebih cepat, seperti pada saat daya pancar berada pada level -17.1 dBm, waktu satu kali transmisi untuk kooperatif *network coding* adalah 3.8 detik, sedangkan untuk kooperatif tanpa *network coding* adalah 4.3 detik.
4. Skema *network coding* pada sistem komunikasi kooperatif selain memberikan keuntungan pada waktu satu kali transmisi juga memberikan efek pada *throughput* yang dihasilkan, seperti pada saat daya pancar berada pada level -17.1 dBm, *throughput* yang dihasilkan oleh komunikasi kooperatif dengan skema *network coding* 2.6 kbps, sedangkan pada kooperatif tanpa *network coding* adalah 2.2 kbps.

5.2 Saran

Sebagai pengembangan untuk penelitian selanjutnya, data yang dikirim dapat dikembangkan berupa konten multimedia (gambar, suara, text), untuk pengembangan implementasi perangkat yang digunakan bisa lebih dari tiga supaya realisasi terlihat semakin real seperti sistem komunikasi yang sudah ada.



DAFTAR PUSTAKA

- [1] Hong, Y.-W.P., Huang, W.-J., Kuo, C.-C.J., 2010. Cooperative Communications and Networking: Technologies and System Design. Springer Science & Business Media.
- [2] N. Aria, T. E. Hunter, and A. Hedayat, "Cooperative Communication in Wireless Network," Oct-2004.
- [3] Zhang, L., Cimini, L.J., 2008. Cooperative network coding in selective decode-and-forward networks with multiple source-destination pairs, in: 42nd Annual Conference on Information Sciences and Systems, 2008. CISS 2008. Presented at the 42nd Annual Conference on Information Sciences and Systems, 2008. CISS 2008, pp. 521–526. doi:10.1109/CISS.2008.4558581
- [4] Yani, Kalfika. 2012. *Software Defined Radio (SDR)*. (Online), (http://www.len.co.id/len_web/artikel_detail/?mg%3D%3D, diakses 1 Juli 2015).
- [5] Murphy, Patrick., Ashu Sabharwal., Behnaam Aazhang. *Design of WARP: A Wireless Open-Access Research Platform*. Department of Electrical and Computer Engineering, Rice University.
- [6] miqbal.staff.telkomuniversity.ac.id/?p=344 (Online), (diakses 10 Juni 2015, Jam 21.05)
- [7] T. Ho and D. Lun, *Network Coding: An Introduction*. Cambridge University Press, 2008.
- [8] Soung Chang Liew, Shengli Zhang, " Physical Layer Network Coding : Tutorial, Survey And Beyond", Physical Communication 6 4-42, 2013.
- [9] Su,W., Sadek,A.K., and Liu,K. J. R., "Cooperative Communication Protocols in Wireless Networks", New Orleans, 2007
- [10] Patrick Murphy, Ashu Sabharwal, and Behnaam Aazhang, "Design of WARP: a Wireless Open-Access Research Platform," 14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006.
- [11] Sklar, Berhard, " Digital Communications Fundamentals and Applications". Prentice-Hall.2011
- [12] Rhee, Man Young. " Error Correctiong Coding Theory", McGraw-Hill, 1989.

- [13] Lin, Shu. Daniel j. Costello, JR., “ Error Control Coding Fundamentals and Applications”. Prentice-Hall.1983.
- [14] Michael Knox, and Elza Erkip, “Implementation Of Cooperative Communication Using Software Defined Radios,” IEEE, October 2010
- [15] <http://warp.rice.edu/trac/>
- [16] Maxim Integrated Product Inc. MAX2828/2829 Single-/Dual-Band 802.11a/b/g World-band Transceiver ICs Datasheet. 2004.
- [17] H, Dereje, Woldegebreal, Holger Karl. Network Coding based Adaptive Decode and Forward Cooperative Transmission in a Wireless Network : Outage Analysis, in proceeding of the 13th European wireless conference. April 2007

LAMPIRAN A : PENGESAHAN TUGAS AKHIR

Jurusan Teknik Elektro
Fakultas Teknologi Industri – ITS

TE 141599 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Diefa Agung Riyadi
Nomor Pokok : 2213106007
Bidang Studi : Telekomunikasi Multimedia
Tugas Diberikan : Semester Gasal 2015/2016
Dosen Pembimbing : 1. Dr. Ir. Suwadi, M.T.
2. Dr. Ir. Wirawan, DEA.


16 SEP 2015

Judul Tugas Akhir : **Implementasi dan Analisa Kinerja Sistem Komunikasi Kooperatif *Decoding and Forward* (DF) dengan *Network Coding* Menggunakan *Wireless Open-Access Research Platform* (WARP)**
Implementation and Performance Analysis of Decoding and Forward (DF) Cooperative Communication System with Network Coding Using Wireless Open-Access Research Platform (WARP)


Uraian Tugas Akhir :

Network coding adalah teknik dimana data yang dikirimkan akan dikodekan dan diterjemahkan dengan tujuan untuk meningkatkan *throughput*, mengurangi *delay* dan membangun jaringan yang kuat. Sistem komunikasi kooperatif merupakan komunikasi yang memanfaatkan relay yang berfungsi untuk mengeliminasi pengaruh fading pada kanal nirkabel sehingga dapat meningkatkan kinerja sistem, skema *decoding and forward* yaitu mengkodekan sinyal dari *source* sebelum sampai ke *destination*, penggunaan *network coding* pada sistem komunikasi kooperatif dimaksudkan untuk meningkatkan kinerjanya salah satunya pada *throughput* yang dihasilkan. *Wireless Open-Access Research Platform* (WARP) merupakan salah satu *Software Define Radio* (SDR) yang berfungsi untuk mengimplementasikan sistem komunikasi kooperatif dengan *network coding*. Pada tugas akhir ini *network coding* akan dikombinasikan dengan sistem komunikasi kooperatif *decoding and forward* yang kemudian akan disimulasikan pada Matlab dan diimplementasikan pada WARP, kemudian akan dilakukan pengukuran dengan *bit error rate* (BER) sebagai parameternya yang akan digunakan untuk menganalisa seberapa baik kinerja sistem komunikasi kooperatif yang dikombinasikan dengan *network coding*.


Dosen Pembimbing I


Dr. Ir. Suwadi, M.T.
NIP. 19680818 1993031 002


Mengetahui
Jurusan Teknik Elektro FTI – ITS
Ketua,


Dr. Tri Anief Sardjono, S.T., M.T.
NIP. 19700212 1995121 001

Dosen Pembimbing II


Dr. Ir. Wirawan, DEA.
NIP. 19631109 1989031 011

Menyetujui
Bidang Studi Telekomunikasi Multimedia
Koordinator,


Dr. Ir. Endroyono, DEA.
NIP. 19650404 1991021 001

LAMPIRAN B : DOKUMENTASI PENGUKURAN



Pengukuran Indoor



Pengukuran Outdoor



LAMPIRAN C : LISTING PROGRAM

```
%% BER Calculation Process

clear all;
close all;
clc;

warning off;

%% Initiation and Defining Parameters

frame_tx;

ber_timer = tic;

save_file = 0;
    % 0 for simulation
    % 1 for WARP Implementation

select_mode = 0;
    % 0 for simulation
    % 1 for WARP Implementation

n
    = 3;

% Parameters
if(select_mode)
    fc          = 2.4;
    channel     = 11;
    TxGainBB   = 1;      % Tx Baseband Gain. In [0:3]
    TxGainRF   = 0:n:60; % Tx RF Gain. In [0:63]
    RxGainBB   = 3;      % Rx Baseband Gain in [0:31]
    RxGainRF   = 1;      % Rx RF Gain in [1:3]
    gains      = TxGainRF;
else
    % bit to noise ratio
    EbNodB     = [1:n:30];
    gains      = 1:length(EbNodB);
end

% statistical iteration
stat_iter     = 1;

% experiment data set container
temp_dat_set1 = zeros(stat_iter,length(gains));
temp_dat_set2 = zeros(stat_iter,length(gains));
```

```

temp_dat_set3 = zeros(stat_iter,length(gains));
temp_dat_set4 = zeros(stat_iter,length(gains));

% temporary toc data
temp_toc_data1 = zeros(stat_iter,length(gains));
temp_toc_data2 = zeros(stat_iter,length(gains));

% temporary throughput data
temp_thronc_data = zeros(stat_iter,length(gains));
temp_throtc_data = zeros(stat_iter,length(gains));

% BER Container
BER_SISO = zeros(size(gains));
BER_Multi = zeros(size(gains));
BER_NC1 = zeros(size(gains));
BER_NC2 = zeros(size(gains));
BER_Coop = zeros(size(gains));

% Throughput Container
Thro_NC = zeros(size(gains));
Thro_TC = zeros(size(gains));

nn_iter = 1;

for iteration = 1:stat_iter
    for gain = 1:length(gains)
        %% Cooperative DF NC
        tic;

        if(select_mode)
            [rx_vec_air_SR1,rx_vec_air_SD1] =
warp_transmit(tx_vec_air_SD1,TxGainBB,TxGainRF(gain),RxGainB
B,RxGainRF,fc,channel);
        else
            rx_vec_air_SR1 =
awgn(tx_vec_air_SD1,EbNodB(gain));
            rx_vec_air_SD1 =
awgn(tx_vec_air_SD1,EbNodB(gain));
        end

        if(select_mode)
            [rx_vec_air_SR2,rx_vec_air_SD2] =
warp_transmit1(tx_vec_air_SD2,TxGainBB,TxGainRF(gain),RxGain
BB,RxGainRF,fc,channel);
        else
            rx_vec_air_SR2 =
awgn(tx_vec_air_SD2,EbNodB(gain));
            rx_vec_air_SD2 =
awgn(tx_vec_air_SD2,EbNodB(gain));

```

```

end

frame_relay;

if(select_mode)
    [rx_vec_air_RS1,rx_vec_air_RD1] =
warp_transmit2(tx_vec_air_RD,TxGainBB,TxGainRF(gain),RxGainB
B,RxGainRF,fc,channel);
else
    rx_vec_air_RS1 =
awgn(tx_vec_air_RD,EbNodB(gain));
    rx_vec_air_RD1 =
awgn(tx_vec_air_RD,EbNodB(gain));
end

frame_rx1;
frame_rx2;

combining;
tnc = toc;
temp_toc_data1(iteration,gain) = toc;
%
fprintf('\nDone. Process time = %f s \n',pr_time);
% Count BER
[N1,B1] = biterr(bit_sl,xor_SC);
BER_NC = B1;

% Count Throughput
K_NC = (length(bit_sl) - biterr(bit_sl,xor_SC));
Thro_NC = (K_NC/tnc)*(1 - BER_NC);

%
save each statistical iteration data
temp_dat_set1(iteration,gain) = BER_NC;
temp_thronc_data(iteration,gain) = Thro_NC;

clear sym_SD1 sym_RD1
fprintf('Done for NC with Gain = %d\n',gain*n);
end
nn_iter = nn_iter+1;
end
time_nc_mean = mean (temp_toc_data1,1);
thro_nc_mean = mean (temp_thronc_data,1); % in bps
thro_nc_mean = thro_nc_mean/1e3; % convert to kbps

```



```

%% frame_tx.m

clear all;
close all;
clc;

warning off;

% Proses pembentukan frame untuk proses transmisi data. Pada
% tiap-tiap langkah akan dijabarkan proses pembentukan frame.

%Data bit
% definisi data biner yang ditransmisikan. Data bisa
% berbentuk apapun,
% misalnya, biner acak/ data dari suara/gambar/video

%% Init

% Proses inisialisasi sistem. Hal yang dilakukan adalah
% sebagai berikut:
% - Definisi vektor waktu
% - Definisi Frekuensi sampling sistem
% - Definisi sistem modulasi

% Frequency Sampling and Time Sampling
Fs = 40e6;
Ts = 1/Fs;

% Time Vector
L = 2^14;
t = (0:L-1)/Fs;

% Modulation that Used
M = 4;
k = log2(M);
if M > 2
    ph_off = 0;
    h_bb_mod = modem.pskmod('M',M,'PhaseOffset',ph_off);
    h_bb_demod = modem.pskdemod('M',M,'PhaseOffset',ph_off);
else
    h_bb_mod = modem.pskmod('M',M);
    h_bb_demod = modem.pskdemod('M',M);
end

%%

% Random Binary
n_bits = 16384;

```

```

% Generating random binary
rand('seed',1)
bit_s1 = randi([0 1],n_bits,1);
bit_s2 = randi([0 1],n_bits,1);

n_syms = n_bits;

% Encode Data
% Encoding message using convolutional code
t = poly2trellis([3],[5 7]);
codeword_s1 = convenc(bit_s1,t);
coderate = 1/2;
bitSource_s1 = codeword_s1;
n_sym1 = reshape(bitSource_s1,n_syms,k);
n_dat1 = bi2de(n_sym1,'left-msb');

t = poly2trellis([3],[5 7]);
codeword_s2 = convenc(bit_s2,t);
coderate = 1/2;
bitSource_s2 = codeword_s2;
n_sym2 = reshape(bitSource_s2,n_syms,k);
n_dat2 = bi2de(n_sym2,'left-msb');

%Hitung total data payload yang dibutuhkan, setelah
dikurangi preamble,
%txDelay dan CSI bits
%Tentukan symbol rate, cari nilai upsample rate untuk SRRC
Filter. Satuan
%symbol rate adalah symbol/s
Rsym = 5e6;
nsamp = floor(Fs/Rsym);
%Nilai nsamp harus integer, setelah pembulatan kebawah,
hitung rate simbol
%sebenarnya.
Rsym = Fs/nsamp;

%Pembentukan Preamble
% Preamble bertujuan untuk penentuan titik sampling pertama
pada penerima.
%pream = [-1;-1;-1;1;-1;0;0;0;0;0;0;0;0];
pream = [0;0;0;0;-1+1i;1-1i;-1+1i;-1+1i;-1+1i;1-1i;-
1+1i;0;0;0;0;0;0;0;0];
pream_len = length(pream);

%Correlation Matrix
% Pembentukan matriks korelasi preamble
corr_win = 300; %at least corr_win >= pream_len*nsamp

```

```

preamp_up = upsample(pream,nsamp);
ref_vect = zeros(corr_win,1);
ref_vect(1:length(pream_up)) = pream_up;
ref_matx = toeplitz(ref_vect,circshift(ref_vect(corr_win:-
1:1),1));

%Definisi Transmission delay
% Panjang frame L akan dikurangi sejumlah delay untuk
menangani
% transmission delay. Jika semua frame terisi informasi,
ketika terjadi
% delay pada proses transmisi, maka akan ada data yang tidak
tertangkap
% pada penerima. Sehingga diperlukan delay transmisi. Satuan
: sampel
txDelay = 400;

%Phase ambiguity detector/ Channel State Information
sequence
% Bertujuan untuk menangani ambiguitas fasa/perputaran pada
konstellasi.
% Agar sistem SISO bisa berfungsi dengan baik, data yang
diterima harus
% terletak pada kuadran yang sesuai. Gunakan CSI dengan
nilai 1
CSI_syms = 150; %Jumlah simbol untuk CSI
CSI_dat = ones(CSI_syms,1);

%Desain Square Root Raised Cosine
%Parameter desain menggunakan nsamp sebagai upsample rate,
N_SRRC sebagai
%orde filter dan roll-off factor R_SRRC
shape = 'Square Root Raised Cosine';
N_Ord = 6; %Genap
R_SRRC = 0.9; %[0-1] Rolloff Factor
SRRC_spec =
fdesign.pulseshaping(nsamp,shape,'Nsym,Beta',N_Ord,R_SRRC);
h_SRRC = design(SRRC_spec);
flt_delay = h_SRRC.order/2; %or nsamp*N_Ord
norm_factor = max(h_SRRC.numerator);
h_SRRC.numerator = h_SRRC.numerator/norm_factor;
%Normalisasi filter

%Jumlah simbol maksimum yang bisa dimuat dalam satu
transmission frame,
%data_sym mencakup CSI dan data yang ditransmisikan
data_sym = floor((L-(pream_len*nsamp)-txDelay-
flt_delay)/nsamp);

```

```

%Baseband Modulation
% Mulai proses modulasi baseband (BB). Kecuali preamble dan
CSI, semua
% symbol frame akan mengalami modulasi baseband.
sym1 = modulate(h_bb_mod,n_dat1);
sym2 = modulate(h_bb_mod,n_dat2);

%Bit stuffing and frame construction
% Bit stuffing adalah memasukan symbol 1 (CSI) secara
periodik dalam 1
% frame transmisi. Fungsi bit stuffing ini adalah sebagai
metode untuk
% melakukan interpolasi dan estimasi kanal pada penerima.
Dalam satu frame
% transmisi akan memiliki total simbol <= data_sym
tergantung dari jumlah
% symbol CSI yang digunakan.
% Representasi grafis bit stuffing:
% |-----data_sym-----|
% -----|
% CSI - <sub-frames> -- CSI - <sub-frames> -- ... - <sub-
frames> -- CSI

%Tentukan jumlah simbol data diantara simbol CSI
sym_size = floor(data_sym/(CSI_syms-1))-1;
%Tentukan total sub-frame data
sub_frame = CSI_syms-1;
%Total panjang frame aktual dengan stuffed bits
fr_len = sub_frame*sym_size + CSI_syms;
%Update nilai data_sym
data_sym = fr_len - CSI_syms;
%Hitung total frame data, hitung padding, tambahkan bit
stuffing
n_frames = ceil(n_syms/data_sym);
stuff_sym = [upsample(CSI_dat(1:CSI_syms-
1),sym_size+1);CSI_dat(end)];
n_dat_fr = repmat(stuff_sym,1,n_frames);
pad = n_frames*data_sym - n_syms;
%Tambahkan zero padding pada symbol
sym1 = [sym1;zeros(pad,1)];
sym2 = [sym2;zeros(pad,1)];
%Reshape n_sym_dat
sym1 = reshape(sym1,data_sym,n_frames);           % Sebelum
di kirim ke WARP
sym2 = reshape(sym2,data_sym,n_frames);           % Sebelum
di kirim ke WARP

%Bagi data menjadi beberapa frame, tambahkan bit stuffing
for ii=1:n_frames

```

```

    n_dat_fr1(:,ii) =
sym_stuffing(n_dat_fr(:,ii),sym1(:,ii),sym_size);
    n_dat_fr2(:,ii) =
sym_stuffing(n_dat_fr(:,ii),sym2(:,ii),sym_size);
end

%Tambahkan preamble
temp_mat1 = repmat(pream,1,n_frames);
xt_dat1 = [temp_mat1;n_dat_fr1];
xt_dat2 = [temp_mat1;n_dat_fr2];

%Interpolasi, matched filtering, tambahkan filter delay
xt_mf_bb1 =
filter(h_SRRC,[upsample(xt_dat1,nsamp);zeros(flt_delay,n_frames)]);
xt_mf1 = zeros(size(xt_mf_bb1));

xt_mf_bb2 =
filter(h_SRRC,[upsample(xt_dat2,nsamp);zeros(flt_delay,n_frames)]);
xt_mf2 = zeros(size(xt_mf_bb2));

%Clear unused variables
% clear temp_mat1 stuff_sym n_sym

%Upconvert semua frame, normalisasi
time1 = [(0:length(xt_mf1)-1)/Fs].';
time2 = [(0:length(xt_mf2)-1)/Fs].';
f = 5e6;
for ii=1:n_frames
    xt_mf1(:,ii) = xt_mf_bb1(:,ii).*exp(1i*2*pi*f*time1);
    norm_fact1 = max( [ max(real(xt_mf1(:,ii))),
max(imag(xt_mf1(:,ii))) ] );
    %Normalize frame
    tx_vec_air_SD1(:,ii) = xt_mf1(:,ii)./norm_fact1; %% yang
dikirim ke WARP

    xt_mf2(:,ii) = xt_mf_bb2(:,ii).*exp(1i*2*pi*f*time2);
    norm_fact2 = max( [ max(real(xt_mf2(:,ii))),
max(imag(xt_mf2(:,ii))) ] );
    %Normalize frame
    tx_vec_air_SD2(:,ii) = xt_mf2(:,ii)./norm_fact2; %% yang
dikirim ke WARP
end

```

```

%% frame_relay.m
% File ini akan menjalankan proses demodulasi hasil
transmisi pada WARP.
% Proses demodulasi meliputi:
% - Sinkronisasi
% - Downconvert
% - Matched Filtering
% - Timing recovery (Preamble)
% - Baseband demodulation
% Setelah proses baseband modulation, skema SISO dilakukan
pada
% simbol-simbol yang diterima pada kanal yang berbeda.
Kemudian proses hard
% detection dilakukan,
% lakukan proses network coding
% Memodulasikan data setelah proses network coding dst

%% Sinkronisasi dan Downconvert semua frame

time1 = ((0:length(rx_vec_air_SR1)-1)/Fs).';
rx_IQ_displ = rx_vec_air_SR1;

time2 = ((0:length(rx_vec_air_SR2)-1)/Fs).';
rx_IQ_displ2 = rx_vec_air_SR2;

for ii=1:n_frames
    %vco_sig = syn_frame(rx_IQ(:,ii),f,Fs);
    vco_sig1 = exp(1i*2*pi*f*time1);
    rx_vec_air_SR1(:,ii) =
    rx_vec_air_SR1(:,ii).*conj(vco_sig1);

    vco_sig2 = exp(1i*2*pi*f*time2);
    rx_vec_air_SR2(:,ii) =
    rx_vec_air_SR2(:,ii).*conj(vco_sig2);
end

%% Phase Synchronize
hrx_SRRC = design(SRRC_spec);
hrx_SRRC.numerator = hrx_SRRC.numerator *
(norm_factor*nsamp);

%% Matched Filtering
yt_mf1 =
filter(hrx_SRRC,[rx_vec_air_SR1;zeros(flt_delay,n_frames)]);
yt_mf2 =
filter(hrx_SRRC,[rx_vec_air_SR2;zeros(flt_delay,n_frames)]);

```

```

yt_rx_data = zeros(size(n_dat_fr));

z_sym1 = zeros(size(sym1));
z_sym2 = zeros(size(sym2));

ch_est = zeros(size(n_dat_fr));

%% Symbol Detection for Each frame
for ii=1:n_frames
    temp_vct1 = yt_mf1(:,ii);
    temp_vct2 = yt_mf2(:,ii);
    %   multip = max([ max(real(temp_vct1(1:corr_win))) ...
    %               max(imag(temp_vct1(1:corr_win)))]);
    %   temp_vct1(1:corr_win) = temp_vct1(1:corr_win)./multip;

    %% Find the higher correlation
    correlation1 = abs( (temp_vct1(1:corr_win)') * ref_matx
    );
    pream_start1 = find(correlation1 == max(correlation1));
    first_sample1 = pream_start1+length(pream_up);

    correlation2 = abs( (temp_vct2(1:corr_win)') * ref_matx
    );
    pream_start2 = find(correlation2 == max(correlation2));
    first_sample2 = pream_start2+length(pream_up);

    %% Downsample

    temp_vct11 = temp_vct1(first_sample1:end);
    temp_vct22 = temp_vct2(first_sample2:end);
    %Determine Maximum likelihood first optimum sampling
    point
    %ML_samp = ML_first_sym(temp_vct2,nsamp);
    %Start sampling at initial point

    %% Start Downsample

    temp_vct111 = downsample(temp_vct11(1:end),nsamp);
    temp_vct222 = downsample(temp_vct22(1:end),nsamp);

    %% Trim

    temp_vct1111 = temp_vct111(1:fr_len);
    temp_vct2222 = temp_vct222(1:fr_len);

    %% Saved Trim Vector

```

```

yt_rx_data1(:,ii) = temp_vct1111;
yt_rx_data2(:,ii) = temp_vct2222;

%% Channel Estimate
ch_est1(:,ii) =
estimate_chan(yt_rx_data1(:,ii),sym_size);
ch_est2(:,ii) =
estimate_chan(yt_rx_data2(:,ii),sym_size);

%% Received Symbol
r_sym1 = yt_rx_data1(:,ii).*conj(ch_est1(:,ii));
r_sym2 = yt_rx_data2(:,ii).*conj(ch_est2(:,ii));

multipl = [abs(ch_est1(:,ii)).^2];
multip2 = [abs(ch_est2(:,ii)).^2];

yt_rx_data1(:,ii) = (r_sym1)./multipl;
yt_rx_data2(:,ii) = (r_sym2)./multip2;

%% Symbol Destuffing
sym_stuf1 = sym_destuffing(yt_rx_data1(:,ii),sym_size);
sym_stuf2 = sym_destuffing(yt_rx_data2(:,ii),sym_size);

%% Cut all of CSI results and Collect it to selected
matrixs
z_sym1(:,ii) = sym_stuf1;
z_sym2(:,ii) = sym_stuf2;
end

%% Change yt_rx to vector, cut yt_syms
sym_SR1 = reshape(z_sym1,data_sym*n_frames,1);
sym_SR1 = sym_SR1(1:length(n_dat1)); %%% received
data on relay
sym_SR2 = reshape(z_sym2,data_sym*n_frames,1);
sym_SR2 = sym_SR2(1:length(n_dat2)); %%% received
data on relay

%% Demodulation Process
z_sym1 = demodulate(h_bb_demod,sym_SR1);
z_sym2 = demodulate(h_bb_demod,sym_SR2);

```



```

%% Potong z_sym agar memiliki panjang sama dengan n_dat

z_sym1 = z_sym1(1:length(n_dat1));
rx_siso1= z_sym1;

z_sym2 = z_sym2(1:length(n_dat2));
rx_siso2= z_sym2;

%% Decoding Process

yy1= de2bi(rx_siso1,'left-msb');
rx_bit1=reshape(yy1,1,[]);
tb = 3;
bit_SR1 = vitdec(rx_bit1,t,tb,'trunc','hard');
bit_SR1 = bit_SR1';

yy2= de2bi(rx_siso2,'left-msb');
rx_bit2=reshape(yy2,1,[]);
tb = 3;
bit_SR2 = vitdec(rx_bit2,t,tb,'trunc','hard');
bit_SR2 = bit_SR2';

%% Network Coding Process

bit_SRrel = xor(bit_SR1,bit_SR2);

%% Retransmit data from relay.....

%% Encode data XOR on relay

t = poly2trellis([3],[5 7]);
codewordRelay = convenc(bit_SRrel,t);
coderate = 1/2;
bit_SR = codewordRelay;

%% Modulate data

n_symSR = reshape(bit_SR,length(codewordRelay)/k,k);
n_datSR = bi2de(n_symSR,'left-msb');

symSR = modulate(h_bb_mod,n_datSR);

%% Add Zero Padding

symSR = [symSR;zeros(pad,1)];

symSR = reshape(symSR,data_sym,n_frames); % Sebelum dikirim
ke WARP, yg dikirim

%% Dividing data into some frames

```

```

for ii=1:n_frames
    n_dat_fr(:,ii) =
sym_stuffing(n_dat_fr(:,ii),symSR(:,ii),sym_size);
end

%% Add Preamble

xt_dat = [temp_mat1;n_dat_fr];

%% Interpolation, match filtering and add filter delay

xt_mf_bb =
filter(h_SRRC,[upsample(xt_dat,nsamp);zeros(flt_delay,n_frames)
]);
xt_mf = zeros(size(xt_mf_bb));

%% Upconvert all frame and normalization

time = [(0:length(xt_mf)-1)/Fs].';
f = 5e6;
for ii=1:n_frames
    xt_mf(:,ii) = xt_mf_bb(:,ii).*exp(1i*2*pi*f*time);
    norm_fact = max( [ max(real(xt_mf(:,ii))) ,
max(imag(xt_mf(:,ii))) ] );
    %Normalize frame
    tx_vec_air_RD(:,ii) = xt_mf(:,ii)./norm_fact; %% yang
dikirim ke WARP
end

```

```

%% FrameRX1.m
%% Jalur SISO
%Sinkronisasi dan Downconvert semua frame
time1 = ((0:length(rx_vec_air_SD1)-1)/Fs).';
rx_vec_air_SD1 = rx_vec_air_SD1;

time2 = ((0:length(rx_vec_air_SD2)-1)/Fs).';
rx_vec_air_SD2 = rx_vec_air_SD2;

for ii=1:n_frames
    %vco_sig = syn_frame(rx_IQ(:,ii),f,Fs);
    vco_sig1 = exp(1i*2*pi*f*time1);
    rx_vec_air_SD1(:,ii) =
rx_vec_air_SD1(:,ii).*conj(vco_sig1);

    vco_sig2 = exp(1i*2*pi*f*time2);
    rx_vec_air_SD2(:,ii) =
rx_vec_air_SD2(:,ii).*conj(vco_sig2);
end

%Sinkronisasi fasa
hrx_SSRC = design(SSRC_spec);
hrx_SSRC.numerator = hrx_SSRC.numerator *
(norm_factor*nsamp);

%Matched filtering
yt_mf1 =
filter(hrx_SSRC,[rx_vec_air_SD1;zeros(flt_delay,n_frames)]);
yt_mf2 =
filter(hrx_SSRC,[rx_vec_air_SD2;zeros(flt_delay,n_frames)]);

yt_rx_data = zeros(size(n_dat_fr));

z_sym1 = zeros(size(sym1));
z_sym2 = zeros(size(sym2));

ch_est = zeros(size(n_dat_fr));

%Deteksi simbol pertama pada tiap-tiap frame
for ii=1:n_frames
    temp_vct1 = yt_mf1(:,ii);
    temp_vct2 = yt_mf2(:,ii);

    %Cari korelasi tertinggi
    correlation1 = abs( (temp_vct1(1:corr_win')) * ref_matx
);
    pream_start1 = find(correlation1 == max(correlation1));
    first_sample1 = pream_start1+length(pream_up);

```

```

correlation2 = abs( (temp_vct2(1:corr_win)') * ref_matx
);
pream_start2 = find(correlation2 == max(correlation2));
first_sample2 = pream_start2+length(pream_up);

%Downsampling
temp_vct11 = temp_vct1(first_sample1:end);
temp_vct22 = temp_vct2(first_sample2:end);

%Start sampling at initial point
temp_vct111 = downsample(temp_vct11(1:end),nsamp);
temp_vct222 = downsample(temp_vct22(1:end),nsamp);

%Trim
temp_vct1111 = temp_vct111(1:fr_len);
temp_vct2222 = temp_vct222(1:fr_len);

%Save trimmed vector
yt_rx_data1(:,ii) = temp_vct1111;
yt_rx_data2(:,ii) = temp_vct2222;

%Estimasi kanal
ch_est1(:,ii) =
estimate_chan(yt_rx_data1(:,ii),sym_size);
ch_est2(:,ii) =
estimate_chan(yt_rx_data2(:,ii),sym_size);

%Received symbols:
r_sym1 = yt_rx_data1(:,ii).*conj(ch_est1(:,ii));
r_sym2 = yt_rx_data2(:,ii).*conj(ch_est2(:,ii));

multip1 = [abs(ch_est1(:,ii)).^2];
multip2 = [abs(ch_est2(:,ii)).^2];

yt_rx_data1(:,ii) = (r_sym1)./multip1;
yt_rx_data2(:,ii) = (r_sym2)./multip2;

%Symbol de-stuffing
sym_stuf1 = sym_destuffing(yt_rx_data1(:,ii),sym_size);
sym_stuf2 = sym_destuffing(yt_rx_data2(:,ii),sym_size);

%Potong semua frame CSI, kumpulkan hasil data pada
matriks tertentu
z_sym1(:,ii) = sym_stuf1;
z_sym2(:,ii) = sym_stuf2;
end

%Ubah yt_rx_data menjadi vektor, potong yt_syms

```

```

sym_SD1 = reshape(z_sym1,data_sym*n_frames,1);
sym_SD1 = sym_SD1(1:length(n_dat1)); %% data yang diterima
di destination1

sym_SD2 = reshape(z_sym2,data_sym*n_frames,1);
sym_SD2 = sym_SD2(1:length(n_dat2)); %% data yang diterima
di destination2

%%
%Demodulasi
z_sym1 = demodulate(h_bb_demod,sym_SD1);
z_sym2 = demodulate(h_bb_demod,sym_SD2);

%Potong z_sym agar memiliki panjang sama dengan n_dat
z_sym1 = z_sym1(1:length(n_dat1));
rx_siso1= z_sym1;

z_sym2 = z_sym2(1:length(n_dat2));
rx_siso2= z_sym2;

yy1= de2bi(rx_siso1,'left-msb');
rx_bit1=reshape(yy1,1,[]);
tb = 3;
bit_SD1 = vitdec(rx_bit1,t,tb,'trunc','hard');
bit_SD1 = bit_SD1';

yy2= de2bi(rx_siso2,'left-msb');
rx_bit2=reshape(yy2,1,[]);
tb = 3;
bit_SD2 = vitdec(rx_bit2,t,tb,'trunc','hard');
bit_SD2 = bit_SD2';

```

```

%%FRAME_DEMOD.m
%% Jalur Relay ke destination

%Sinkronisasi dan Downconvert semua frame
time1 = ((0:length(rx_vec_air_RS1)-1)/Fs).';
rx_vec_air_RS1 = rx_vec_air_RS1;

time2 = ((0:length(rx_vec_air_RD1)-1)/Fs).';
rx_vec_air_RD1 = rx_vec_air_RD1;

for ii=1:n_frames
    %vco_sig = syn_frame(rx_IQ(:,ii),f,Fs);
    vco_sig1 = exp(1i*2*pi*f*time1);
    rx_vec_air_RS1(:,ii) =
    rx_vec_air_RS1(:,ii).*conj(vco_sig1);

    vco_sig2 = exp(1i*2*pi*f*time2);
    rx_vec_air_RD1(:,ii) =
    rx_vec_air_RD1(:,ii).*conj(vco_sig2);
end

%Sinkronisasi fasa
hrx_SSRC = design(SRRC_spec);
hrx_SSRC.numerator = hrx_SSRC.numerator *
(norm_factor*nsamp);

%Matched filtering
yt_mf1 =
filter(hrx_SSRC,[rx_vec_air_RS1;zeros(flt_delay,n_frames)]);
yt_mf2 =
filter(hrx_SSRC,[rx_vec_air_RD1;zeros(flt_delay,n_frames)]);

yt_rx_data = zeros(size(n_dat_fr));

z_sym1 = zeros(size(symSR));
z_sym2 = zeros(size(symSR));

ch_est = zeros(size(n_dat_fr));

%Deteksi simbol pertama pada tiap-tiap frame
for ii=1:n_frames
    temp_vct1 = yt_mf1(:,ii);
    temp_vct2 = yt_mf2(:,ii);

    %Cari korelasi tertinggi
    correlation1 = abs( (temp_vct1(1:corr_win')) * ref_matx
);
    pream_start1 = find(correlation1 == max(correlation1));
    first_sample1 = pream_start1+length(pream_up);

```

```

correlation2 = abs( (temp_vct2(1:corr_win)') * ref_matx
);
pream_start2 = find(correlation2 == max(correlation2));
first_sample2 = pream_start2+length(pream_up);

%Downsampling
temp_vct2_1 = temp_vct1(first_sample1:end);
temp_vct2_2 = temp_vct2(first_sample2:end);

temp_vct3_1 = downsample(temp_vct2_1(1:end),nsamp);
temp_vct3_2 = downsample(temp_vct2_2(1:end),nsamp);

%Trim
temp_vct3_1 = temp_vct3_1(1:fr_len);
temp_vct3_2 = temp_vct3_2(1:fr_len);

%Save trimmed vector
yt_rx_data1(:,ii) = temp_vct3_1;
yt_rx_data2(:,ii) = temp_vct3_2;

%Estimasi kanal
ch_est1(:,ii) =
estimate_chan(yt_rx_data1(:,ii),sym_size);
ch_est2(:,ii) =
estimate_chan(yt_rx_data2(:,ii),sym_size);

%Received symbols:
r_sym1 = yt_rx_data1(:,ii).*conj(ch_est1(:,ii));
r_sym2 = yt_rx_data2(:,ii).*conj(ch_est2(:,ii));

multip1 = [abs(ch_est1(:,ii)).^2];
multip2 = [abs(ch_est2(:,ii)).^2];

yt_rx_data1(:,ii) = (r_sym1)./multip1;
yt_rx_data2(:,ii) = (r_sym2)./multip2;

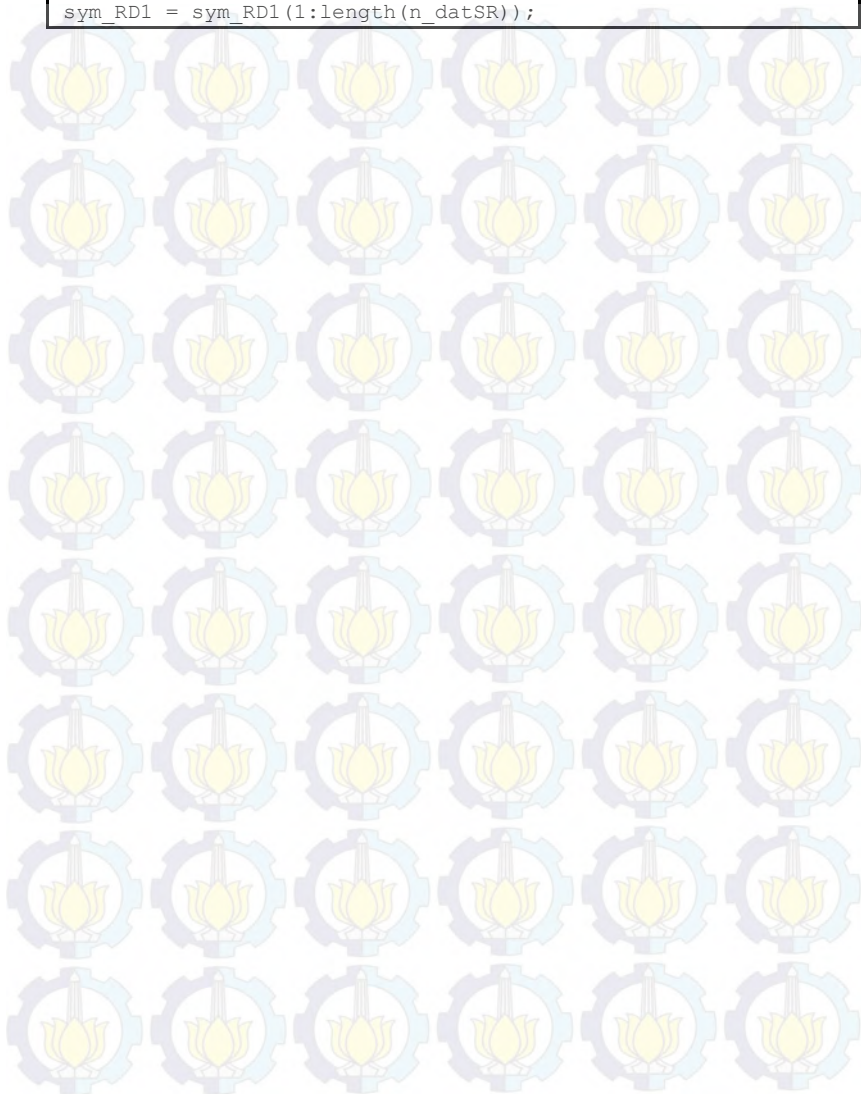
sym_stuf1 = sym_destuffing(yt_rx_data1(:,ii),sym_size);
sym_stuf2 = sym_destuffing(yt_rx_data2(:,ii),sym_size);

%Potong semua frame CSI, kumpulkan hasil data pada
matriks tertentu
z_sym1(:,ii) = sym_stuf1;
z_sym2(:,ii) = sym_stuf2;
end

%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_RS1 = reshape(z_sym1,data_sym*n_frames,1);
sym_RS1 = sym_RS1(1:length(n_datSR));

```

```
sym_RD1 = reshape(z_sym2,data_sym*n_frames,1);  
sym_RD1 = sym_RD1(1:length(n_datSR));
```




```

function [rx_vec_air_SR2,rx_vec_air_SD2] =
warp_transmit1(tx_vec_air_SD2,TxGainBB,TxGainRF,RxGainBB,RxGainRF,fc,channel)

%WLABV7_TXXR_FRAMES Summary of this function goes here
% Detailed explanation goes here
NUMNODES = 3;

% Create a vector of node objects
nodes = wl_initNodes(NUMNODES);

USE_AGC = false;

%Create a UDP broadcast trigger and tell each node to be ready for it
eth_trig = wl_trigger_eth_udp_broadcast;
wl_triggerManagerCmd(nodes,'add_ethernet_trigger',[eth_trig]);

%Get IDs for the interfaces on the boards. Since this example assumes each
%board has the same interface capabilities, we only need to get the IDs
%from one of the boards
[RFA,RFB] = wl_getInterfaceIDs(nodes(3));

%Set up the interface for the experiment
wl_interfaceCmd(nodes,'RF_ALL','tx_gains',TxGainBB,TxGainRF);
wl_interfaceCmd(nodes,'RF_ALL','channel',fc,channel);

if(USE_AGC)
    wl_interfaceCmd(nodes,'RF_ALL','rx_gain_mode','automatic');
    wl_basebandCmd(nodes,'agc_target',-10);
    wl_basebandCmd(nodes,'agc_trig_delay', 511);
else
    wl_interfaceCmd(nodes,'RF_ALL','rx_gain_mode','manual');
    % RxGainRF = 2; % Rx RF Gain in [1:3]
    % RxGainBB = 12; % Rx Baseband Gain in [0:31]

    wl_interfaceCmd(nodes,'RF_ALL','rx_gains',RxGainRF,RxGainBB);
end

%We'll use the transmitter's I/Q buffer size to determine how long our

```

```

%transmission can be
txLength = nodes(3).baseband.txIQLen;

%Set up the baseband for the experiment
wl_basebandCmd(nodes, 'tx_delay', 0);
wl_basebandCmd(nodes, 'tx_length', txLength);

node_tx      = nodes(3);
node_rx_1    = nodes(2);
node_rx_2    = nodes(1);
RF_TX = RFA;
RF_RX = RFA;

Ts = 1/(wl_basebandCmd(nodes(3), 'tx_buff_clk_freq'));
Ts_RSSI = 1/(wl_basebandCmd(nodes(3), 'rx_rssi_clk_freq'));

%enabling both tx and rx buffer
%Write to buffer and enable both tx, rx

tx_vec_air_SD2_size = size(tx_vec_air_SD2);
rx_vec_air_SR2 = zeros(txLength, tx_vec_air_SD2_size(2));
rx_vec_air_SD2 = zeros(txLength, tx_vec_air_SD2_size(2));
rx_RSSI = zeros(txLength/4, tx_vec_air_SD2_size(2));
rx_gains = zeros(3, tx_vec_air_SD2_size(2));

%Do transmit-receive for all st frames
wl_interfaceCmd(node_tx, RF_TX, 'tx_en');
wl_interfaceCmd(node_rx_1, RF_RX, 'rx_en');
wl_interfaceCmd(node_rx_2, RF_RX, 'rx_en');

wl_basebandCmd(node_tx, RF_TX, 'tx_buff_en');
wl_basebandCmd(node_rx_1, RF_RX, 'rx_buff_en');
wl_basebandCmd(node_rx_2, RF_RX, 'rx_buff_en');

for m=1:tx_vec_air_SD2_size(2);
    wl_basebandCmd(node_tx, [RF_TX], 'write_IQ',
    tx_vec_air_SD2(:,m) );
    %send trigger
    eth_trig.send();
    %Get received data

    %First RF
    rx_vec_air_SR2(:,m) =
    wl_basebandCmd(node_rx_1, [RF_RX], 'read_IQ', 0, txLength);
    rx_vec_air_SD2(:,m) =
    wl_basebandCmd(node_rx_2, [RF_RX], 'read_IQ', 0, txLength);

```

```
rx_RSSI(:,m) =  
wl_basebandCmd(node_rx_1,[RF_RX],'read_RSSI',0,txLength/(Ts_  
RSSI/Ts));  
rx_RSSI(:,m) =  
wl_basebandCmd(node_rx_2,[RF_RX],'read_RSSI',0,txLength/(Ts_  
RSSI/Ts));  
  
rx_gains(:,m) =  
wl_basebandCmd(node_rx_1,[RF_RX],'agc_state');  
rx_gains(:,m) =  
wl_basebandCmd(node_rx_2,[RF_RX],'agc_state');  
end  
  
%Disable all RF to both node  
wl_basebandCmd(nodes,'RF_ALL','tx_rx_buff_dis');  
wl_interfaceCmd(nodes,'RF_ALL','tx_rx_dis');  
end
```

```

function [rx_vec_air_RS1,rx_vec_air_RD1] =
warp_transmit2(tx_vec_air_RD,TxGainBB,TxGainRF,RxGainBB,RxGainRF,fc,channel)

%WLABV7 TXRX_FRAMES Summary of this function goes here
% Detailed explanation goes here
NUMNODES = 3;

% Create a vector of node objects
nodes = wl_initNodes(NUMNODES);

USE_AGC = false;

%Create a UDP broadcast trigger and tell each node to be
ready for it
eth_trig = wl_trigger_eth_udp_broadcast;
wl_triggerManagerCmd(nodes,'add_ethernet_trigger',[eth_trig]
);

%Get IDs for the interfaces on the boards. Since this
example assumes each
%board has the same interface capabilities, we only need to
get the IDs
%from one of the boards
[RFA,RFB] = wl_getInterfaceIDs(nodes(2));

%Set up the interface for the experiment
wl_interfaceCmd(nodes,'RF_ALL','tx_gains',TxGainBB,TxGainRF)
;
wl_interfaceCmd(nodes,'RF_ALL','channel',fc,channel);

if(USE_AGC)

wl_interfaceCmd(nodes,'RF_ALL','rx_gain_mode','automatic');
wl_basebandCmd(nodes,'agc_target',-10);
wl_basebandCmd(nodes,'agc_trig_delay', 511);
else
wl_interfaceCmd(nodes,'RF_ALL','rx_gain_mode','manual');
% RxGainRF = 2; % Rx RF Gain in [1:3]
% RxGainBB = 12; % Rx Baseband Gain in [0:31]

wl_interfaceCmd(nodes,'RF_ALL','rx_gains',RxGainRF,RxGainBB)
;
end

```

```

%We'll use the transmitter's I/Q buffer size to determine
how long our
%transmission can be
txLength = nodes(2).baseband.txIQLen;

%Set up the baseband for the experiment
wl_basebandCmd(nodes,'tx_delay',0);
wl_basebandCmd(nodes,'tx_length',txLength);

node_tx      = nodes(2);
node_rx_1    = nodes(1);
node_rx_2    = nodes(3);
RF_TX = RFA;
RF_RX = RFA;

Ts = 1/(wl_basebandCmd(nodes(1),'tx_buff_clk_freq'));
Ts_RSSI = 1/(wl_basebandCmd(nodes(1),'rx_rssi_clk_freq'));

%enabling both tx and rx buffer
%Write to buffer and enable both tx, rx

tx_vec_air_RD_size = size(tx_vec_air_RD);
rx_vec_air_RS1 = zeros(txLength,tx_vec_air_RD_size(2));
rx_vec_air_RD1 = zeros(txLength,tx_vec_air_RD_size(2));
rx_RSSI = zeros(txLength/4,tx_vec_air_RD_size(2));
rx_gains = zeros(3,tx_vec_air_RD_size(2));

%Do transmit-receive for all st frames
wl_interfaceCmd(node_tx,RF_TX,'tx_en');
wl_interfaceCmd(node_rx_1,RF_RX,'rx_en');
wl_interfaceCmd(node_rx_2,RF_RX,'rx_en');

wl_basebandCmd(node_tx,RF_TX,'tx_buff_en');
wl_basebandCmd(node_rx_1,RF_RX,'rx_buff_en');
wl_basebandCmd(node_rx_2,RF_RX,'rx_buff_en');

for m=1:tx_vec_air_RD_size(2);
    wl_basebandCmd(node_tx,[RF_TX], 'write_IQ',
tx_vec_air_RD(:,m) );
    %send trigger
    eth_trig.send();
    %Get received data

    %First RF
    rx_vec_air_RS1(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX],'read_IQ', 0, txLength);
    rx_vec_air_RD1(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX],'read_IQ', 0, txLength);

```

```
rx_RSSI(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX],'read_RSSI',0,txLength/(Ts_
RSSI/Ts));
rx_RSSI(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX],'read_RSSI',0,txLength/(Ts_
RSSI/Ts));

rx_gains(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX],'agc_state');
rx_gains(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX],'agc_state');
end

%Disable all RF to both node
wl_basebandCmd(nodes,'RF_ALL','tx_rx_buff_dis');
wl_interfaceCmd(nodes,'RF_ALL','tx_rx_dis');
end
```

```

%% Initial
close all;
clear all;
clc;

warning off;

M = 4; %modulation of orde PSK
nbit = 1e5; %numbers of binary information
snr = 0:30;
time = 1.25e-6;

err_SC1 = []; ber_SC1 = [];
err_MRC1 = []; ber_MRC1 = [];
err_SCNC1 = []; ber_SCNC1 = [];
err_MRCNC1 = []; ber_MRCNC1 = [];
err_SISO = []; ber_SISO = [];
err_Multi = []; ber_Multi = [];

%% TX (Timeslot 1)
bit_s1 = randint(nbit,1,[0 1]);
% bit_s1 = [1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 0];

% Encode
t = poly2trellis([3],[5 7]);
codeword1 = convenc(bit_s1,t);
coderate = 1/2;

% modulator
cw1 = reshape(codeword1,[],2);
cwDec1 = bi2de(cw1,'left-msb');
sym1 = pskmod(cwDec1,M);

% channel
h = 1;
rxSign1 = conv(sym1,h);

%% TX (Timeslot 2)
bit_s2 = randint(nbit,1,[0 1]);
% bit_s2 = [1 0 1 1 0 0 1 1 1 1 0 1 1 0 1 0];

% Encode
t = poly2trellis([3],[5 7]);
codeword2 = convenc(bit_s2,t);
coderate = 1/2;

```

```

% modulator
cw2 = reshape(codeword2,[],2);
cwDec2 = bi2de(cw2,'left-msb');
sym2 = pskmod(cwDec2,M);

% channel
h = 1;
rxSign2 = conv(sym2,h);
%% RX

lenSign1 = length(rxSign1);
lenSign2 = length(rxSign2);

for b = 1:length(snr)
    %% S-R and R-D path (Information 1)
    tic;
    % Relay receive signal from source (S-R)
    vSR1 =
    1/sqrt(2)*(randn(lenSign1,1)+i*(randn(lenSign1,1)));
    nSR1 = (10.^(-snr(b)./20)).*vSR1;
    rxSignSR1 = rxSign1 + nSR1;

    % demodulate on relay
    decSR1 = pskdemod(rxSignSR1,M);
    cwSR1 = de2bi(decSR1,'left-msb');
    cwSR1 = reshape(cwSR1,[],1);

    % decode on relay
    tb = 2;
    bitEstSR1 = vitdec(cwSR1,t,tb,'trunc','hard');

    %% S-R and R-D path (Information 2)
    % Relay receive signal from source (S-R)
    vSR2 =
    1/sqrt(2)*(randn(lenSign2,1)+i*(randn(lenSign2,1)));
    nSR2 = (10.^(-snr(b)./20)).*vSR2;
    rxSignSR2 = rxSign2 + nSR2;

    % demodulate on relay
    decSR2 = pskdemod(rxSignSR2,M);
    cwSR2 = de2bi(decSR2,'left-msb');
    cwSR2 = reshape(cwSR2,[],1);

    % decode on relay
    tb = 2;
    bitEstSR2 = vitdec(cwSR2,t,tb,'trunc','hard');

```



```

%% Relay Process

% XOR between data 1 and data 2
bitRelay = xor(bitEstSR1,bitEstSR2);

% Encode XOR Data
codewordRelay = convenc(bitRelay,t);
coderate = 1/2;

% modulate XOR data
cw_R = reshape(codewordRelay,[],2);
cwDec_R = bi2de(cw_R,'left-msb');
sym_R = pskmod(cwDec_R,M);

% Destination receive broadcast XOR Data R-D1,D2
rxSignRD = conv(sym_R,h);
vRD =
1/sqrt(2)*(randn(lenSign1,1)+i*(randn(lenSign1,1)));
nRD = (10.^(-snr(b)./20)).*vRD;
rxSignRD = rxSignRD + nRD;

%% S-D Path (SISO Data)

% Destination receive direct information from source 1
vSDNC1 =
1/sqrt(2)*(randn(lenSign1,1)+i*(randn(lenSign1,1)));
nSDNC1 = (10.^(-snr(b)./20)).*vSDNC1;
rxSignSDNC1 = rxSign1 + nSDNC1;
rxSignSDNC1 = rxSignSDNC1 + nRD;

% demodulator on destination 2
decSDNC1 = pskdemod(rxSignSDNC1, M);
cwSDNC1 = de2bi(decSDNC1,'left-msb');
cwSDNC1 = reshape(cwSDNC1,[],1);

% decoder on destination 2
tb = 2;
bitEstSDNC1 = vitdec(cwSDNC1,t,tb,'trunc','hard');

% Destination receive direct information from source 2
vSDNC2 =
1/sqrt(2)*(randn(lenSign2,1)+i*(randn(lenSign2,1)));
nSDNC2 = (10.^(-snr(b)./20)).*vSDNC2;
rxSignSDNC2 = rxSign2 + nSDNC2;
rxSignSDNC2 = rxSignSDNC2 + nRD;

% demodulator on destination 1
decSDNC2 = pskdemod(rxSignSDNC2, M);
cwSDNC2 = de2bi(decSDNC2,'left-msb');

```

```

cwSDNC2 = reshape(cwSDNC2,[],1);

% decoder on destination 1
tb = 2;
bitEstSDNC2 = vitdec(cwSDNC2,t,tb,'trunc','hard');

%% Cooperative System with NC

% Selective Combining
mag_rx_SDNC1 = abs(rxSignSDNC1);
mag_rx_RDNC = abs(rxSignRD);

if mag_rx_SDNC1 > mag_rx_RDNC
    SCNC1 = rxSignSDNC1;
else
    SCNC1 = rxSignRD;
end

% demodulator on destination with selective combining
decSCNC1 = pskdemod(SCNC1,M);
codeSCNC1 = de2bi(decSCNC1,'left-msb');
codeSCNC1 = reshape(codeSCNC1,[],1);

% decode on destination
tb = 2;
bitEstSCNC1 = vitdec(codeSCNC1,t,tb,'trunc','hard');

% data on dest 2 SC
bitEstDestSCNC2 = xor(bit_s2,bitEstSCNC1);

% Selective Combining
mag_rx_SDNC2 = abs(rxSignSDNC2);
mag_rx_RDNC = abs(rxSignRD);

if mag_rx_SDNC2 > mag_rx_RDNC
    SCNC2 = rxSignSDNC2;
else
    SCNC2 = rxSignRD;
end

% demodulator on destination with selective combining
decSCNC2 = pskdemod(SCNC2,M);
codeSCNC2 = de2bi(decSCNC2,'left-msb');
codeSCNC2 = reshape(codeSCNC2,[],1);

% decode on destination
tb = 2;
bitEstSCNC2 = vitdec(codeSCNC2,t,tb,'trunc','hard');

```

```

% data on dest 1 SC
bitEstDestSCNC1 = xor(bit_s1,bitEstSCNC2);

toc;
tnc = toc;

err_SDNC_SC1 = sum(abs(bitEstDestSCNC2-bit_s1));
berSDNC_SC1(b) = err_SDNC_SC1/nbit;
throSDNC_SC1 = (length(bit_s1) -
err_SDNC_SC1)./(time*3*length(bit_s1));

err_SCNC1 = [err_SCNC1 biterr(bit_s1,bitEstDestSCNC2)];
ber_SCNC1 = [ber_SCNC1
biterr(bit_s1,bitEstDestSCNC2)/length(bit_s1)];
thro_SCNC1 = (length(bit_s1) -
err_SCNC1)./(time*3*length(bit_s1));

% Max. Ratio Combining
MRCNC1 = mag_rx_SDNC1.*rxSignSDNC1 +
mag_rx_RDNC.*rxSignSR1;
MRCNC1 = MRCNC1 - (mag_rx_SDNC1.*rxSignSDNC1);

% demodulator on destination with Max. Ratio Combining
decMRCNC1 = pskdemod(MRCNC1,M);
codeMRCNC1 = de2bi(decMRCNC1,'left-msb');
codeMRCNC1 = reshape(codeMRCNC1,[],1);

% decoder on destination
tb = 2;
bitEstMRCNC1 = vitdec(codeMRCNC1,t,tb,'trunc','hard');

% data on destination 2 with MRC
bitEstDestMRCNC2 = xor(bit_s2,bitEstMRCNC1);

err_SDNC_MRC1 = sum(abs(bitEstMRCNC1-bit_s1));
berSDNC_MRC1(b) = err_SDNC_MRC1/nbit;
throSDNC_MRC1 = (length(bit_s1) -
err_SDNC_MRC1)./(time*3*length(bit_s1));

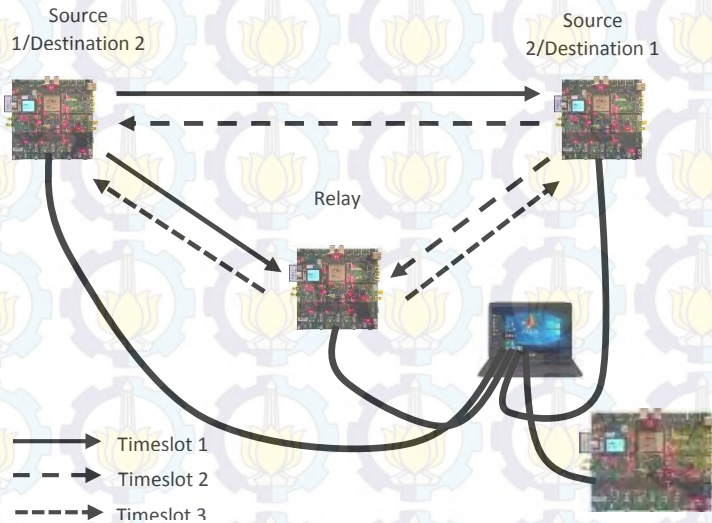
err_MRCNC1 = [err_MRCNC1 biterr(bit_s1,bitEstMRCNC1)];
ber_MRCNC1 = [ber_MRCNC1
biterr(bit_s1,bitEstMRCNC1)/length(bit_s1)];
thro_MRCNC1 = (length(bit_s1) -
err_MRCNC1)./(time*3*length(bit_s1));

end

```

LAMPIRAN D : SETUP WARP

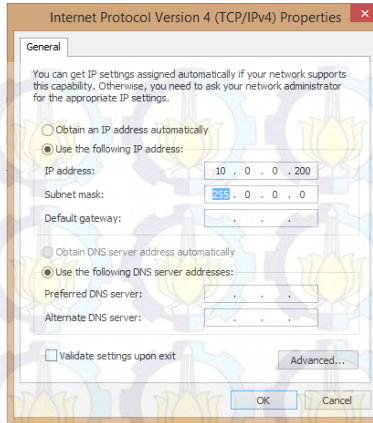
Berikut adalah cara integrasi (*setup*) Laptop/PC dalam hal ini yang menjalankan MATLAB dengan hardware *Wireless Open-Access Research Platform (WARP)*



Integrasi MATLAB ke hardware *Wireless Open-Access Research Platform (WARP)* untuk Sistem Komunikasi Kooperatif dengan Skema *Network Coding*

Adapun langkah untuk mengintegrasikan WARP dengan PC/Laptop adalah sebagai berikut :

1. Menggabungkan modul WARP dan PC/Laptop menggunakan *switch Ethernet* dan kabel LAN seperti pada gambar.
2. Mengatur IP Lokal yang ada pada PC/Laptop seperti pada gambar berikut



Pengaturan IP Address pada PC/Laptop

3. Mengatur IP *address* pada WARP, karena WARP yang digunakan adalah tiga buah, maka IP yang dipakai, 10.0.0.1 untuk node 1, 10.0.0.2 untuk node 2, 10.0.0.3 untuk node 3, pengaturan dilakukan dengan mengubah *dipswitch* yang ada pada modul WARP, seperti pada gambar



Konfigurasi *dipswitch* Untuk Mengatur IP pada WARP

4. Memastikan setiap node sudah terhubung ke *ethernet switch* dengan cara tes koneksi ping ke setiap node
ping 10.0.0.1 ke node 1
ping 10.0.0.2 ke node 2
ping 10.0.0.3 ke node 3

- Mengisi tabel routing lewat *command prompt* dengan mengetikkan perintah “arp -s 10.0.0.255 ff-ff-ff-ff-ff-ff”, kemudian memeriksa tabel routing dengan mengetikkan “arp -a”

```

C:\Windows\system32>arp -a

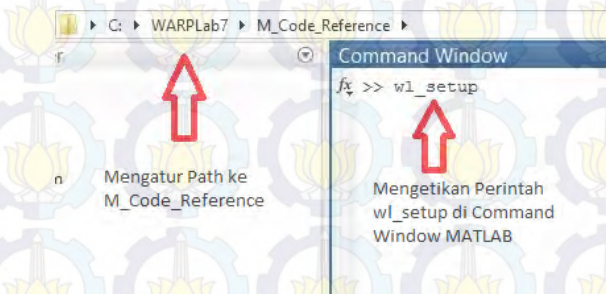
Interface: 10.0.0.200 --- 0x3
Internet Address      Physical Address      Type
10.255.255.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250      01-00-5e-7f-ff-fa    static

C:\Windows\system32>arp -s 10.0.0.255 ff-ff-ff-ff-ff-ff
C:\Windows\system32>arp -a

Interface: 10.0.0.200 --- 0x3
Internet Address      Physical Address      Type
10.0.0.255            ff-ff-ff-ff-ff-ff    static
10.255.255.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250      01-00-5e-7f-ff-fa    static
  
```

Pengaturan Tabel Routing

- Mengatur pengaturan WARP pada MATLAB dengan cara, mengarahkan direktori MATLAB ke “M-Code Reference” WARPLab 7, kemudian ketikkan perintah “wl_setup” pada *command window* MATLAB



Set Path Folder WARPLab pada MATLAB

- Memastikan koneksi modul WARP dengan PC/Laptop dengan mengetikkan perintah “wl_initNodes(3)” pada *command window*, angka dalam kurung menyatakan bahwa ada 3 nodes yang digunakan, untuk indikator keberhasilannya dapat dilihat pada Gambar

Displaying properties of 3 wl_node objects:

ID	WLVER	HWVER	Serial #	Ethernet MAC Addr	Address
0	7.1.0	2	N/A	00-50-C2-63-30-00	10.0.0.1
1	7.1.0	2	N/A	00-50-C2-63-30-01	10.0.0.2
2	7.1.0	2	N/A	00-50-C2-63-30-02	10.0.0.3

>>

Konfigurasi *Nodes* pada MATLAB

BIOGRAFI PENULIS



Penulis bernama Diefa Agung Riyadi adalah anak dari Bapak Darsono dan Ibu Sarinah, dilahirkan di Jakarta, 10 April 1992. Penulis menyelesaikan pendidikan dasar di SDN Ciracas 10 Pagi Jakarta pada tahun 2004, selepas pendidikan dasar penulis melanjutkan pendidikan menengah di SMP Negeri 174 SSN Jakarta yang diselesaikan pada tahun 2007, setelah itu penulis melanjutkan pendidikan menengah berikutnya di SMA Negeri 58 Jakarta yang diselesaikan pada tahun 2010. Penulis kemudian melanjutkan ke jenjang Diploma 3 di Politeknik Negeri Jakarta dengan mengambil jurusan Teknik Elektro program studi Teknik Telekomunikasi yang diselesaikan pada tahun 2013, tidak berhenti sampai disitu penulis melanjutkan pendidikan ke jenjang Sarjana pada tahun 2014 di Institut Teknologi Sepuluh Nopember, Surabaya dengan Jurusan Teknik Elektro melalui program S1 Lintas Jalur periode genap dan memilih konsentrasi pada bidang studi Telekomunikasi Multimedia. Penulis bisa dihubungi melalui alamat riyadidief@gmail.com