



TUGAS AKHIR - TE 141599

**IMPLEMENTASI DAN ANALISIS KINERJA SISTEM
KOMUNIKASI KOOPERATIF *AMPLIFY AND
FORWARD* DENGAN SKEMA *NETWORK CODING*
PADA MODUL WARP (*WIRELESS OPEN-ACCESS
RESEARCH PLATFORM*)**

Ummul Khair
NRP. 2213106010

Dosen Pembimbing
Dr. Ir. Suwadi, MT.
Dr. Ir. Wirawan, DEA.

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



FINAL PROJECT - TE 141599

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

Ummul Khair
NRP. 2213106010

Supervisors
Dr. Ir. Suwadi, MT.
Dr. Ir. Wirawan, DEA.

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

**IMPLEMENTASI DAN ANALISIS KINERJA SISTEM
KOMUNIKASI KOOPERATIF *AMPLIFY AND
FORWARD* DENGAN SKEMA *NETWORK CODING*
PADA MODUL WARP (*WIRELESS OPEN-ACCESS
RESEARCH PLATFORM*)**

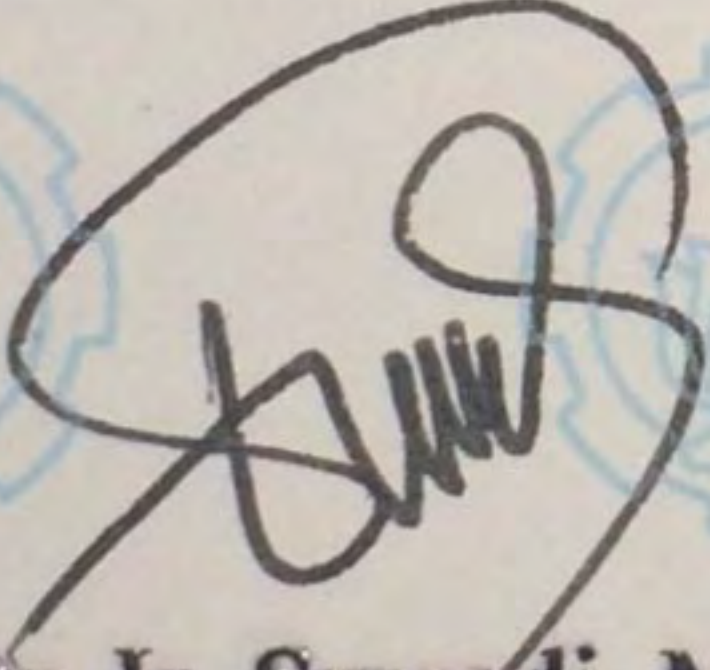
TUGAS AKHIR

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

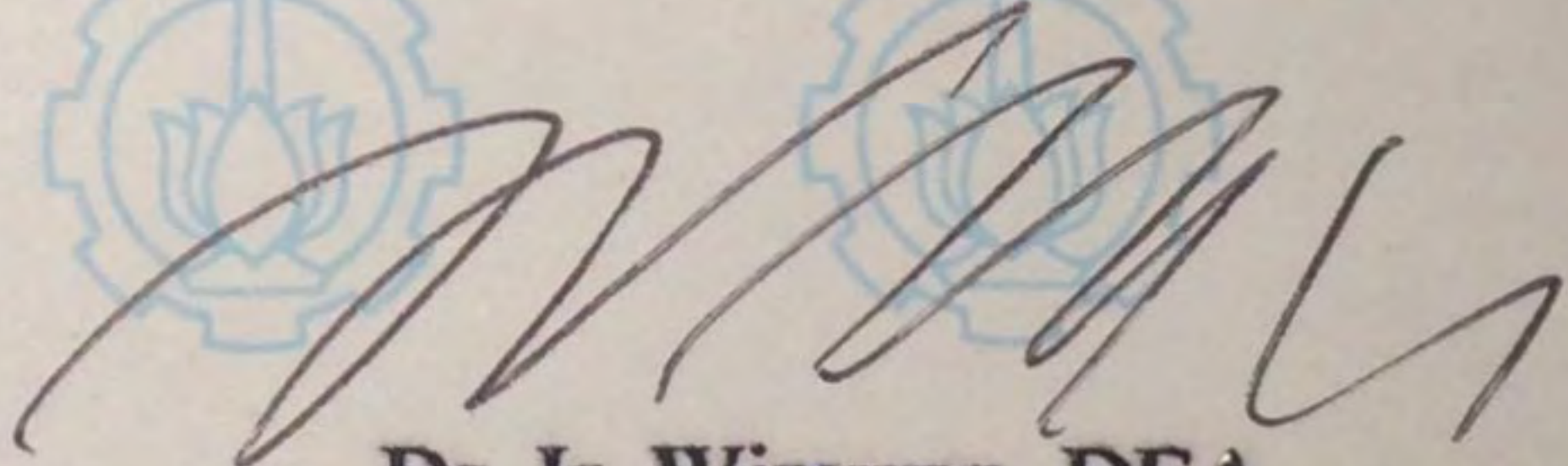
Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II


Dr. Ir. Suwadi, MT,

NIP. 19680818 1993031 002


Dr. Ir. Wirawan, DEA,

NIP. 19631109 1989031 011



Implementasi dan Analisis Kinerja Sistem Komunikasi Kooperatif *Amplify and Forward* dengan skema *Network Coding* pada modul WARP (*Wireless Open-Access Research Platform*)

Nama : Ummul Khair
Pembimbing : Dr. Ir. Suwadi, MT.
Dr. Ir. Wirawan, DEA

ABSTRAK

Network Coding adalah teknik di mana data yang dikirimkan dikodekan dan diterjemahkan untuk meningkatkan *throughput*, mengurangi keterlambatan dan membangun jaringan yang lebih kuat. *Physical Network Coding* (PNC) pada awalnya diusulkan sebagai cara untuk meningkatkan performa *network coding* menggunakan gelombang elektromagnetik. Komunikasi kooperatif bertujuan untuk mengurangi gangguan *multipath fading*. Skema *amplify* dan *forward* (AF) sistem komunikasi kooperatif adalah informasi yang dikirimkan ke node *relay* akan mengalami penguatan dan dikirimkan ke node *destination*. Pada Tugas Akhir ini akan dilakukan simulasi kinerja dari sistem komunikasi kooperatif AF tanpa dan dengan *network coding* dan akan diimplementasikan pada modul SDR (*Software Defined Radio*) yaitu WARP (*Wireless Open-Access Research Platform*), parameter yang diambil sebagai acuan performansi adalah *bit error rate* (BER). Setelah dilakukan implementasi, sistem komunikasi kooperatif *amplify and forward* dengan skema tradisional memiliki performansi yang lebih bagus dibandingkan *network coding*. Pada saat *power transmit* -27 dBm, untuk skema tradisional BER sudah bernilai 0, sedangkan untuk skema *network coding* pada saat *power transmit* 40 dBm baru BER bernilai 0. Namun di tinjau dari segi *throughput*, dengan skema *network coding* dibandingkan tradisional, skema *network coding* meningkatkan *throughput* sebesar 20% dibandingkan tradisional.

Kata Kunci : Sistem Komunikasi Kooperatif; *Amplify and Forward*; *Physical Network Coding*; WARP



Halaman ini sengaja dikosongkan

Implementation and Performance Analysis Of Amplify and Forward Cooperative Communication System with Network Coding Scheme using WARP (Wireless Open-Access Research Platform) module

Name : Ummul Khair
Supervisors : Dr. Ir. Suwadi, MT.
Dr. Ir. Wirawan, DEA

ABSTRACT

A technique that the sending data is being coded and translated to increase throughput, reduce delays and build stronger network is called Network Coding. Physical Network Coding (PNC) was recommended as a method to increase network coding's performance by using electromagnetic resonance. Cooperative communication aims to reduce interference multipath fading. The scheme of amplify and forward (AF) cooperative communication system is a first node's relay which receiving message from resource node without passing decoding and detection, and directly strengthen and sending to the destination node. In this research will be done a performance simulation of cooperative communication AF system with and without network coding. Also, will be implemented into Software Defined Radio (SDR) module is Wireless Open-Access Research Platform (WARP) that performance reference parameter is the bit error rate (BER). Adding networking coding into cooperative communication system with AF relay hopefully will increase the performance of cooperative communication system. After implementation, the communication system of cooperative amplify and forward the traditional scheme has performasi better than network coding. When the power is -27 dBm transmit, to traditional BER scheme has value 0, while for the network coding scheme when the transmit power of 40 dBm new BER worth 0. However in the review of the terms of throughput, with the coding scheme than traditional network, network coding scheme increases throughput by 20% compared tradisinonal.

Key Words : Cooperative Communication System; Amplify and Forward; Physical Network Coding; WARP



Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala karunia dan bantuan-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul:

“Implementasi dan Analisis Kinerja Sistem Komunikasi Kooperatif Amplify and Forward dengan skema Network Coding pada modul WARP (*Wireless Open-Access Research Platform*)”

Tugas Akhir ini merupakan salah satu syarat yang harus dipenuhi dalam menyelesaikan Program Studi Strata 1 pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Dalam proses pengerjaan Tugas Akhir ini penulis telah terbantu oleh beberapa pihak, antara lain:

1. Kedua orang tua, Zulkarnaini dan Darmiati, Kakak-kakak penulis Dian Maulana, Suci Handani, Aulia Rahim, dan Arsyad Fathon yang selalu mendukung, mendoakan dan memberikan bantuan berupa material dan non material.
2. Bapak Dr. Ir. Suwadi, MT dan Bapak Dr. Ir. Wirawan, DEA selaku dosen pembimbing yang telah memberikan bantuan teknis dan semangat ketika penulis mengalami permasalahan dalam penelitian.
3. Ketua jurusan, Bapak dan Ibu Dosen Pengajar Lintas Jalur Teknik Elektro ITS atas ilmu dan bimbingannya selama proses perkuliahan.
4. Keluarga besar Laboratorium Telekomunikasi Multimedia B304, khususnya tim riset Network Coding : Bapak Muhammad Iqbal, Mbak Anindita Kemala, Prastiyanto, Diefa Agung Riyadi, dan Bambang Eko Surya
5. Teman-teman Lintas Jalur Genap 2013, khususnya program studi Teknik Telekomunikasi Multimedia, Sherly, Danar, Banyu, Mbak Dwi, Chipe, Joko, Mbak Dita, Dessy, Mbak Tania dan Mbak Nita atas kerja samanya selama masa perkuliahan dan menyelesaikan Tugas Akhir.

Surabaya, Januari 2016

Penulis



Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK.....	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Metodologi.....	2
1.5 Sistematika Penulisan.....	4
1.6 Manfaat Penulisan.....	5
BAB II TEORI PENUNJANG.....	7
2.1 Sistem Komunikasi Nirkabel.....	7
2.2 Sistem Komunikasi Kooperatif.....	7
2.3 Skenario Sistem Komunikasi Kooperatif.....	9
2.4 Protokol Relay pada Sistem Kooperatif.....	9
2.4.1. Protokol Relay <i>Amplify And Forward</i>	10
2.4.2. Protokol Relay <i>Decode And Forward</i>	12
2.5 Teknik Combining [8].....	12
2.6 Modulasi QPSK.....	14
2.7 <i>Network Coding</i>	15
2.7.1 Skema Tradisional Tanpa Network Coding (TS).....	16
2.7.2 Skema Straightforward Network Coding (SNC).....	17
2.7.3 Skema Physical Layer Network Coding (PNC).....	17

2.7.4	Analisis Performansi <i>Pseudo Physical Layer Network Coding amplify and forward</i> (PPNCAF) [7]	21
2.8	Software Defined Radio	24
2.9	<i>Wireless Open-Access Research Platform</i> dan WARPLab [6]	25
2.9.1	WARP Board	26
2.9.2	WARPLab	31
BAB III PERANCANGAN DAN IMPLEMENTASI		33
3.1	Alur Perancangan Sistem	33
3.2	Desain Sistem	34
3.3	Integrasi PC dengan Modul WARP	36
3.4	Pemodelan Sistem Komunikasi	36
3.4.1	Parameter Sistem Komunikasi	37
3.4.2	Sistem Komunikasi	37
3.5	Implementasi Secara Realtime pada Modul WARP	38
3.5.1	Pada Pemancar	38
3.5.2	Proses Pengiriman Sinyal dan Komunikasi	40
3.5.3	Proses pada Penerima	41
3.6	Metode Pengukuran dan perhitungan BER dan <i>Throughput</i>	42
3.6.1	Pengukuran Lingkungan <i>Indoor</i>	43
3.5.1	Pengukuran Lingkungan <i>Outdoor</i>	44
3.7	Perhitungan BER	45
3.8	Perhitungan <i>Throughput</i>	45
BAB IV HASIL PENGUKURAN DAN ANALISIS DATA		47
4.1	Analisis Hasil Simulasi Sistem Komunikasi Kooperatif	47
4.2	Output implementasi Sistem Komunikasi di Modul WARP	47
4.3	Hasil Implementasi Sistem pada Modul WARP	50
4.4	Analisis BER Hasil Pengukuran di Lingkungan <i>Indoor</i>	50
4.4.1	Analisa Pengukuran <i>Indoor</i> Kondisi LOS	50

4.4.2	Analisa Pengukuran <i>Indoor</i> Kondisi NLOS.....	57
4.5	Analisis BER Hasil Pengukuran di Lingkungan <i>Outdoor</i>	64
4.5.1.	Analisis Pengukuran <i>Outdoor</i> kondisi LOS	64
4.5.2.	Analisis Pengukuran <i>Outdoor</i> kondisi NLOS	66
4.5.3.	Analisis Pengukuran <i>Outdoor</i> kondisi LOS dan NLOS	68
4.6	Analisis <i>Throughput</i> Hasil Pengukuran.....	69
BAB V PENUTUP.....		73
5.1	Kesimpulan.....	73
5.2	Saran.....	73
DAFTAR PUSTAKA		75
LAMPIRAN A		77
LAMPIRAN B		79
LAMPIRAN B		81
LAMPIRAN C		83
RIWAYAT HIDUP.....		125



Halaman ini sengaja dikosongkan

TABLE OF CONTENT

ABSTRAK	I
ABSTRACT	III
PREFACE	V
CONTENT	VII
ILLUSTRATION	XI
TABLES	XIII
CHAPTER 1 PREFACE	1
1.1 BACKGROUND	1
1.2 PROBLEM FORMULATION	2
1.3 PURPOSES	2
1.4 METHODOLOGY	2
1.5 SYSTEMATICS DISCUSSION	4
1.6 BENEFIT	5
CHAPTER 2 THEORY	7
2.1 WIRELESS COMMUNICATIONS	7
2.2 COOPERATIVE COMMUNICATIONS SYSTEM	7
2.3 SCENARIO OF COOPERATIVE COMMUNICATIONS SYSTEM	9
2.4 RELAY PROTOCOLS	9
2.4.1 Amplify-and-forward relaying protocol	10
2.4.2 Decode and-forward relaying protocol	12
2.5 COMBINING TECHNIQUE	12
2.6 QPSK MODULATION	14
2.7 NETWORK CODING	15
2.7.1 Without Network Coding Scheme	16

2.7.2	Straightforward Network Coding Scheme	17
2.7.3	Physical Layer Network Coding Scheme	17
2.7.4	Pseudo Physical Layer Network Coding amplify forward	21
2.8	SOFTWARE DEFINED RADIO	24
2.9	WIRELESS OPEN-ACCESS RESEARCH PLATFORM	25
2.9.1	WARP Board	26
2.9.2	WARP Lab	31
CHAPTER 3 DESIGN SYSTEM.....		33
3.1	FLOW OF DESIGN SYSTEM.....	31
3.2	DESIGN SYSTEM.....	32
3.3	INTEGRATION PC WITH WARP MODULE.....	36
3.4	MODELING OF COOPERATIVE COMMUNICATION SYSTEM	36
3.4.1	Parameter of Communication System	37
3.4.2	Communication System.....	37
3.5	IMPLEMENTATION AT THE WARP MODULE.....	38
3.5.1	At the Transmitter.....	38
3.5.2	Process Transmission Signal and Communication	40
3.5.3	At the Receiver	41
3.6	SCENARIO OF MEASURING.....	42
3.6.1	Indoor Environment.....	43
3.6.2	Outdoor Environment	44
3.7	BER CALCULATION	45
3.8	THROUGHPUT CALCULATION	45
CHAPTER 4 MEASURING AND DATA ANALYSIS		47
4.1	ANALYSIS OF COOPERATIVE COMMUNICATIONS SYSTEM	47
4.2	OUTPUT IMPLEMENTATION AT THE WARP MODULE	47

4.3	RESULT OF IMPLEMENTATION AT THE WARP MODULE	50
4.4	COMPARISON OF BER IN INDOOR ENVIRONMENT	50
4.4.1	LOS Condition	50
4.4.2	NLOS Condition	57
4.5	COMPARISON OF BER IN OUTDOOR ENVIRONMENT	64
4.5.1	LOS Condition	64
4.5.2	NLOS Condition	66
4.5.3	LOS and NLOS Condition	68
4.6	COMPARISON OF THROUGHPUT	69
	CHAPTER 5 CLOSING.....	73
5.1	CONCLUSION.....	73
5.2	SUGGESTION	73
	REFERENCES	75
	APPENDIX A	77
	APPENDIX B	79
	APPENDIX B.....	81
	APPENDIX C	83
	BIOGRAPHY.....	125



Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK.....	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Metodologi.....	2
1.5 Sistematika Penulisan.....	4
1.6 Manfaat Penulisan.....	5
BAB II TEORI PENUNJANG.....	7
2.1 Sistem Komunikasi Nirkabel.....	7
2.2 Sistem Komunikasi Kooperatif.....	7
2.3 Skenario Sistem Komunikasi Kooperatif.....	9
2.4 Protokol Relay pada Sistem Kooperatif.....	9
2.4.1. Protokol Relay <i>Amplify And Forward</i>	10
2.4.2. Protokol Relay <i>Decode And Forward</i>	12
2.5 Teknik Combining [8].....	12
2.6 Modulasi QPSK.....	14
2.7 <i>Network Coding</i>	15
2.7.1 Skema Tradisional Tanpa Network Coding (TS).....	16
2.7.2 Skema Straightforward Network Coding (SNC).....	17
2.7.3 Skema Physical Layer Network Coding (PNC).....	17

2.7.4	Analisis Performansi <i>Pseudo Physical Layer Network Coding amplify and forward</i> (PPNCAF) [7]	21
2.8	Software Defined Radio	24
2.9	<i>Wireless Open-Access Research Platform</i> dan WARPLab [6]	25
2.9.1	WARP Board	26
2.9.2	WARPLab	31
BAB III PERANCANGAN DAN IMPLEMENTASI		33
3.1	Alur Perancangan Sistem	33
3.2	Desain Sistem	34
3.3	Integrasi PC dengan Modul WARP	36
3.4	Pemodelan Sistem Komunikasi	36
3.4.1	Parameter Sistem Komunikasi	37
3.4.2	Sistem Komunikasi	37
3.5	Implementasi Secara Realtime pada Modul WARP	38
3.5.1	Pada Pemancar	38
3.5.2	Proses Pengiriman Sinyal dan Komunikasi	40
3.5.3	Proses pada Penerima	41
3.6	Metode Pengukuran dan perhitungan BER dan <i>Throughput</i>	42
3.6.1	Pengukuran Lingkungan <i>Indoor</i>	43
3.5.1	Pengukuran Lingkungan <i>Outdoor</i>	44
3.7	Perhitungan BER	45
3.8	Perhitungan <i>Throughput</i>	45
BAB IV HASIL PENGUKURAN DAN ANALISIS DATA		47
4.1	Analisis Hasil Simulasi Sistem Komunikasi Kooperatif	47
4.2	Output implementasi Sistem Komunikasi di Modul WARP	47
4.3	Hasil Implementasi Sistem pada Modul WARP	50
4.4	Analisis BER Hasil Pengukuran di Lingkungan <i>Indoor</i>	50
4.4.1	Analisa Pengukuran <i>Indoor</i> Kondisi LOS	50

4.4.2	Analisa Pengukuran <i>Indoor</i> Kondisi NLOS.....	57
4.5	Analisis BER Hasil Pengukuran di Lingkungan <i>Outdoor</i>	64
4.5.1.	Analisis Pengukuran <i>Outdoor</i> kondisi LOS	64
4.5.2.	Analisis Pengukuran <i>Outdoor</i> kondisi NLOS	66
4.5.3.	Analisis Pengukuran <i>Outdoor</i> kondisi LOS dan NLOS	68
4.6	Analisis <i>Throughput</i> Hasil Pengukuran.....	69
BAB V PENUTUP.....		73
5.1	Kesimpulan.....	73
5.2	Saran.....	73
DAFTAR PUSTAKA		75
LAMPIRAN A		77
LAMPIRAN B		79
LAMPIRAN B		81
LAMPIRAN C		83
RIWAYAT HIDUP.....		125



Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1. 1 Diagram Alir Pengerjaan Tugas Akhir	4
Gambar 2. 1 Model Sistem Komunikasi Kooperatif	8
Gambar 2. 2 Skenario Komunikasi Kooperatif	9
Gambar 2. 3 <i>Amplify-and-Forward</i>	10
Gambar 2. 4 <i>Decode-and-Forward</i>	12
Gambar 2. 5 Konstelasi QPSK pada Matlab	15
Gambar 2. 6 Skema pengiriman pada <i>network coding</i>	16
Gambar 2. 7 <i>Tradisional</i> Tanpa <i>Network Coding</i>	16
Gambar 2. 8 Skema <i>Straightforward Network Coding</i>	17
Gambar 2. 9 Skema Physical Layer Network Coding	18
Gambar 2. 10 <i>System model of PNC</i> [10]	20
Gambar 2. 11 Blok Diagram skema PNC	20
Gambar 2. 12 <i>System Model of PPNCAF</i>	21
Gambar 2. 13 Komponen-komponen utama dari arsitektur WARP	25
Gambar 2. 14 Modul WARP Versi 2.2	26
Gambar 2. 15 Diagram blok node WARP	27
Gambar 2. 16 Radio board	29
Gambar 2. 17 Arsitektur radio board	29
Gambar 2. 18 TX Output vs Gain Settings pada MAX2829 [11]	30
Gambar 2. 19 Gain Rx Baseband pada MAX2829 [11]	31
Gambar 2. 20 Penggunaan Kerangka Kerja WARPLab [6]	32
Gambar 3. 1 Alur Perancangan Sistem	33
Gambar 3. 2 Desain Sistem Komunikasi Kooperatif <i>Network Coding</i>	34
Gambar 3. 3 BER kooperatif tradisional dan <i>network coding</i>	35
Gambar 3. 4 <i>Throughput</i> kooperatif tradisional dan <i>network coding</i>	35
Gambar 3. 5 Integrasi antara PC dengan WARP	36
Gambar 3. 6 Blok sistem komunikasi SIMO pada TS 1 atau 2	37
Gambar 3. 7 Blok diagram Sistem Komunikasi SIMO pada TS 3	38
Gambar 3. 8 Blok <i>Modulator</i> QPSK	39
Gambar 3. 9 Respon impulse filter <i>squared root raised cosine</i>	40
Gambar 3. 10 Proses Pengiriman Sinyal dan Komunikasi	41
Gambar 3. 11 Denah ruang AJ403 untuk pengukuran <i>indoor</i>	43
Gambar 3. 12 kondisi tanpa penghalang	44
Gambar 3. 13 Kondisi dengan penghalang	44
Gambar 3. 14 Denah pengukuran di lingkungan <i>outdoor</i>	45
Gambar 4. 1 Simulasi sistem komunikasi kooperatif	47
Gambar 4. 2 Random Bit Node <i>Source</i> 1 dan dan Node <i>Source</i> 2	48

Gambar 4. 3 Bit yang diterima di Relay setelah tahap <i>Network Coding</i>	48
Gambar 4. 4 Bit yang diterima pada node <i>Destination</i> 1 dan 2	49
Gambar 4. 5 Konstelasi simbol QPSK	49
Gambar 4. 6 BER untuk jalur komunikasi multihop dan SISO	52
Gambar 4. 7 BER perancangan sistem dengan NC dan tanpa NC	52
Gambar 4. 8 BER untuk jalur komunikasi multihop dan SISO	54
Gambar 4. 9 BER perancangan sistem dengan NC dan tanpa NC	54
Gambar 4. 10 BER untuk jalur komunikasi multihop dan SISO	56
Gambar 4. 11 BER perancangan sistem dengan NC dan tanpa NC	56
Gambar 4. 12 BER untuk jalur komunikasi multihop dan SISO	58
Gambar 4. 13 BER perancangan sistem dengan NC dan tanpa NC	59
Gambar 4. 14 BER untuk jalur komunikasi multihop dan SISO	60
Gambar 4. 15 BER perancangan sistem dengan NC dan tanpa NC	61
Gambar 4. 16 BER untuk jalur komunikasi multihop dan SISO	62
Gambar 4. 17 BER perancangan sistem dengan NC dan tanpa NC	63
Gambar 4. 18 BER sistem terhadap fungsi jarak dan Ptx skema NC	65
Gambar 4. 19 BER sistem terhadap fungsi jarak dan Ptx skema NC	67
Gambar 4. 20 BER sistem terhadap fungsi jarak dan Ptx skema NC	69
Gambar 4. 21 <i>Throughput</i> lingkungan <i>indoor</i> kondisi LOS	70
Gambar 4. 22 <i>Throughput</i> lingkungan <i>indoor</i> kondisi NLOS	70
Gambar 4. 23 <i>Throughput</i> lingkungan <i>outdoor</i> kondisi LOS	71
Gambar 4. 24 <i>Throughput</i> lingkungan <i>outdoor</i> kondisi LOS	71

DAFTAR TABEL

Tabel 3. 1 Parameter Sistem Komunikasi	37
Tabel 3. 2 <i>Mapping</i> Modulasi QPSK	39
Tabel 4. 1 BER pada jarak 4 meter (indoor) kondisi LOS	51
Tabel 4. 2 BER pada jarak 6 meter (indoor) kondisi LOS	53
Tabel 4. 3 BER pada jarak 8 meter (indoor) kondisi LOS	55
Tabel 4. 4 BER pada jarak 4 meter (indoor) kondisi NLOS	57
Tabel 4. 5 BER pada jarak 6 meter (indoor) kondisi NLOS	59
Tabel 4. 6 BER pada jarak 8 meter (indoor) kondisi NLOS	61
Tabel 4. 7 BER sistem komunikasi kooperatif dengan skema <i>network coding</i>	64
Tabel 4. 8 BER sistem komunikasi kooperatif dengan skema <i>network coding</i>	66
Tabel 4. 9 BER kondisi LOS dan NLOS pada jarak 4 dan 24 meter	68



DAFTAR TABEL

Tabel 3. 1 Parameter Sistem Komunikasi	37
Tabel 3. 2 <i>Mapping</i> Modulasi QPSK	39
Tabel 4. 1 BER pada jarak 4 meter (indoor) kondisi LOS	51
Tabel 4. 2 BER pada jarak 6 meter (indoor) kondisi LOS	53
Tabel 4. 3 BER pada jarak 8 meter (indoor) kondisi LOS	55
Tabel 4. 4 BER pada jarak 4 meter (indoor) kondisi NLOS	57
Tabel 4. 5 BER pada jarak 6 meter (indoor) kondisi NLOS	59
Tabel 4. 6 BER pada jarak 8 meter (indoor) kondisi NLOS	61
Tabel 4. 7 BER sistem komunikasi kooperatif dengan skema <i>network coding</i>	64
Tabel 4. 8 BER sistem komunikasi kooperatif dengan skema <i>network coding</i>	66
Tabel 4. 9 BER kondisi LOS dan NLOS pada jarak 4 dan 24 meter	68



BAB I

PENDAHULUAN

1.1 Latar Belakang

Komunikasi nirkabel telah mengalami perkembangan selama dua dekade ini. Setiap generasi baru perangkat nirkabel telah membawa perbaikan penting dalam hal *reliability*, *data rates*, *device sizes*, *battery life*, dan *network connectivity* [1]. Kinerja komunikasi nirkabel tidaklah selalu terhindar dari gangguan-gangguan seperti *fading*. Sistem komunikasi kooperatif berfungsi untuk mengeliminasi pengaruh *fading* pada kanal nirkabel, sehingga dapat meningkatkan kinerja sistem [2]. Sistem komunikasi kooperatif adalah sebuah pergeseran paradigma baru untuk generasi keempat (LTE) pada sistem nirkabel yang akan menjamin kecepatan data yang tinggi untuk semua pengguna dalam suatu jaringan. Sistem komunikasi kooperatif ini bisa menjadi kunci aspek teknologi dalam jaringan nirkabel generasi kelima (5G) [3]. *Network coding* menawarkan paradigma baru untuk jaringan komunikasi. Dengan menggunakan *network coding*, maka masalah kecepatan download file, penyimpanan, pesan, komunikasi interaktif, efisiensi dalam distribusi *content* dan masalah tentang *packet loss* dapat diselesaikan, sehingga pengguna internet lebih mudah dalam melakukan akses download file dalam jumlah besar dengan kata lain *network coding* akan meningkatkan *performance* sistem pada jaringan internet [4]. Salah satu metode berbasis *network coding* yang diusulkan untuk mengeksplorasi gelombang EM adalah *Physical Network Coding* (PNC). PNC merupakan satu dari tiga konsep dari *Network Coding* yang awalnya diusulkan sebagai cara untuk meningkatkan performa *network coding* yang terjadi secara alami yang menggunakan gelombang elektromagnetik. Pada PNC bahwa level *symbol* dan *phase* akan di jumlahkan diasumsikan kedatangannya sama dan mengabaikan noise [5]. Pada tugas akhir ini membutuhkan 3 timeslot dalam proses komunikasi. WARP (*Wireless Open-Access Research Platform*) adalah salah satu bentuk *Software Defined Radio* yang dikembangkan sebagai modul penelitian tentang sistem komunikasi nirkabel. WARP merupakan sebuah FPGA (*Field Programmable Gate Array*) atau modul DSP (*Digital Signal Processing*) yang terhubung dengan modul radio [6]. Pada Tugas akhir ini, untuk meningkatkan *performance* sistem komunikasi kooperatif dengan *relay amplify and forward*, maka sistem komunikasi kooperatif ini akan digabungkan dengan *network coding* (PNC) [7] yang akan

diimplementasikan pada modul WARP, setelah itu akan dilakukan pengukuran dan analisis data dengan membandingkan kinerja sistem komunikasi kooperatif dengan dan tanpa *network coding*.

1.2 Perumusan Masalah

Permasalahan yang akan diangkat dalam Tugas Akhir ini adalah :

- a. Bagaimana mengimplementasikan sistem komunikasi kooperatif skema *Physical Network Coding* pada modul WARP?
- b. Bagaimana Pengaruh *physical network coding* terhadap kinerja sistem komunikasi kooperatif AF ?
- c. Membandingkan kinerja dari skema jaringan tanpa *Physical Network Coding* dengan jaringan yang telah menerapkan PNC pada sistem komunikasi kooperatif AF.

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah:

- a. Menerapkan skema *Physical Network Coding* pada sistem komunikasi kooperatif AF di modul WARP.
- b. Mengetahui performansi dari skema *Physical Network Coding* pada sistem komunikasi kooperatif AF
- c. Menganalisis perbedaan antara skema tradisional dan skema *Physical Network Coding*.

1.4 Metodologi

Metodologi yang digunakan pada pengerjaan tugas akhir ini melalui beberapa tahap :

- a. Studi Literatur
Langkah pertama yang dilakukan adalah mengumpulkan literatur yang berhubungan dengan topik tugas akhir, yaitu :
 1. Sistem komunikasi kooperatif
 2. *Network coding*
 3. Simulasi sistem komunikasi
 4. Referensi pemrograman modul WARP
 5. Referensi WARPLab
- b. Perumusan parameter yang digunakan dan metode pengukurannya
Sebelum proses pengukuran bisa berjalan, parameter performansi, yaitu yang menunjukkan kualitas unjuk kerja sistem komunikasi, dirumuskan dan ditentukan metode pengukurannya. Pada tahapan ini, akan ditentukan misalnya, informasi yang akan dipancarkan,

parameter yang diambil sebagai acuan performansi adalah *bit error rate (BER)* dan *throughput*.

c. Simulasi sistem komunikasi

Proses simulasi yaitu dengan menggunakan software Matlab. Simulasi bertujuan untuk melihat apakah algoritma yang digunakan sesuai dengan teori sistem komunikasi.

d. Implementasi pada modul WARP

Sistem komunikasi kooperatif ini akan diimplementasikan pada modul WARP dengan menggunakan software WARPlab

e. Pengukuran data

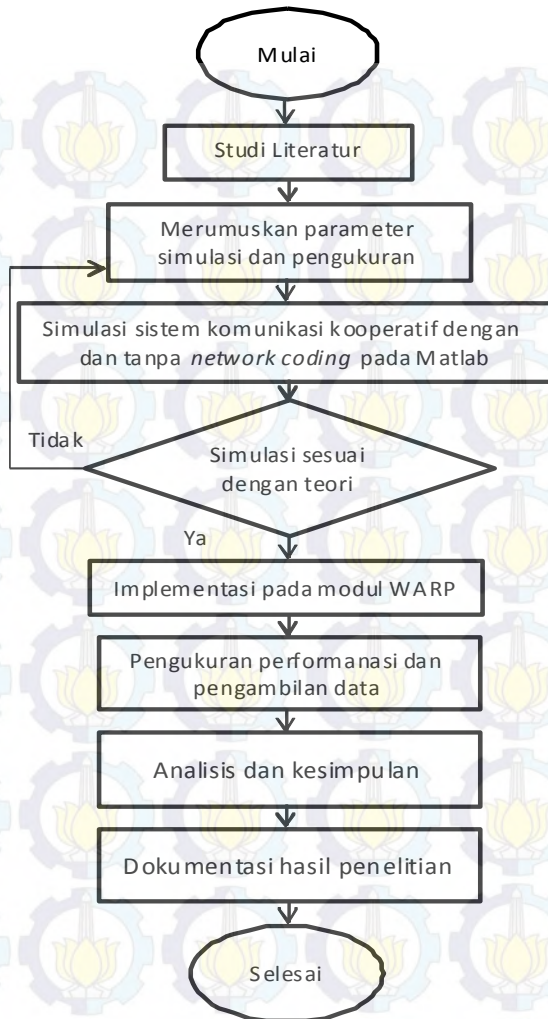
Setelah modul WARP bisa digunakan untuk simulasi dengan kondisi kanal real, maka proses pengukuran dapat dilakukan. Pada tahapan ini, dilakukan pengukuran data untuk berbagai kondisi parameter yang ditentukan, tujuan akhirnya adalah menentukan seberapa baik unjuk kerjanya yang ditunjukkan oleh BER dan *throughput*.

f. Proses Analisis data dan pengambilan kesimpulan

Analisis data dan pengambilan kesimpulan dilakukan setelah pengukuran data selesai.

g. Dokumentasi penelitian

Tahapan akhir ini meliputi pembuatan laporan tugas akhir dan penulisan jurnal ilmiah. Pembuatan laporan tugas akhir untuk beberapa bagian dilakukan bersesuaian dengan pengerjaan tahapan-tahapan diatas. Sedangkan jurnal ilmiah dilakukan setelah laporan tugas akhir selesai. Metodologi yang digunakan pada pengerjaan tugas akhir ini dijabarkan dengan diagram alir sebagai berikut:



Gambar 1. 1 Diagram Alir Pengerjaan Tugas Akhir

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam Tugas Akhir ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini mengandung hal-hal berikut diantaranya latar belakang, perumusan masalah, tujuan tugas akhir, metodologi, sistematika penulisan dan manfaat penulisan

BAB 2 TEORI PENUNJANG

Pada bab ini diuraikan secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada tugas akhir ini. Teori penunjang tugas akhir ini meliputi: sistem komunikasi kooperatif, network coding, SDR (*Software Defined Radio*), *Wireless Open-Access Research Platform* dan WARPLab

BAB 3 METODE PENELITIAN

Bab ini menjelaskan mengenai metode penelitian. Bagian ini berisi proses pembuatan simulasi dan implementasi mulai dari pembangkitan data secara acak, pengolahan data pada tiap node, serta proses perhitungan *Bit Error Rate* (BER) dan *throughput*.

BAB 4 HASIL SIMULASI DAN ANALISIS

Bab ini menjelaskan tentang hasil simulasi yang diperoleh dan analisis kinerja dari hasil simulasi.

BAB 5 PENUTUP

Memberi kesimpulan tentang hasil yang telah diperoleh dan saran yang layak dilakukan bila penelitian ini dilanjutkan.

1.6 Manfaat Penulisan

Secara spesifik manfaat penulisan *Network Coding* untuk sistem komunikasi kooperatif adalah menghasilkan model jaringan nirkabel yang baru yang menjadi kunci aspek teknologi yang baru dalam jaringan nirkabel.



Halaman ini sengaja dikosongkan

BAB II TEORI PENUNJANG

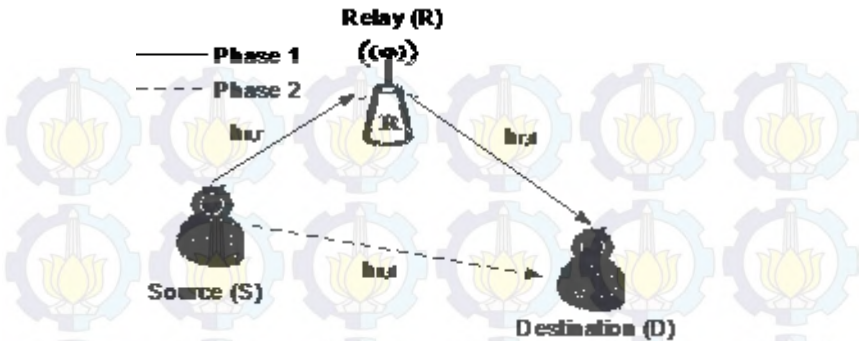
2.1 Sistem Komunikasi Nirkabel

Pada sistem komunikasi nirkabel, ada 2 (dua) jenis transmisi sinyal dalam perambatan gelombangnya yaitu *Line of Sight* (LoS) dan *Non Line of Sight* (NLoS) [2]. *Line of Sight* (LoS) merupakan perambatan gelombang radio dari antena pemancar ke antena penerima tanpa adanya halangan (*obstacle*) karena proses perambatan sinyalnya terjadi secara langsung (*direct path*) dan satu lintasan (*single path*). Sedangkan *Non Line of Sight* (NLoS) adalah perambatan gelombang radio merambat dari antena pemancar ke antena penerima melalui banyak lintasan (*multipath*). Hal ini dikarenakan sinyal radio tersebut melewati *free space* (atmosfir) ataupun penghalang lain seperti pohon, gedung, dan gunung sehingga mengalami *refleksi*, *refraksi* dan *scattering* saat terjadinya proses propagasi sehingga dapat menyebabkan *fading*.

Fading merupakan gangguan pada sinyal yang menyebabkan penurunan daya sinyal yang diterima sehingga kondisi sinyal tidak dapat dikenali lagi (*error*). Dalam hal ini, *rayleigh fading* dimodelkan bila sinyal yang diterima tidak ada unsur sinyal langsung dari pemancar ke penerima atau dengan kata lain sinyal yang dikirimkan bersifat *Non Line of Sight* (NLoS). Disamping itu, jenis fading lain adalah *rician fading* yang dapat dimodelkan bila sinyal yang diterima merupakan sinyal langsung dari pemancar atau dengan kata lain sinyal yang dikirimkan bersifat *Line of Sight* (LoS) [1].

2.2 Sistem Komunikasi Kooperatif

Sistem komunikasi kooperatif merupakan mekanisme yang mengumpulkan sumber daya dari node-node yang terdistribusi untuk meningkatkan seluruh kinerja jaringan nirkabel. Pada sistem komunikasi kooperatif, *source* (S) mengirimkan informasinya secara *broadcast* ke penerima (D) dan ke *relay* (R) yang merupakan *user* lain dalam area yang berada didekatnya. Kemudian sinyal yang diterima oleh R akan diolah terlebih dahulu untuk kemudian dikirimkan ke D. Sinyal yang diterima dari pengirim maupun dari *relay* akan mengalami proses *combining* di penerima [3]. Sistem komunikasi kooperatif ini bertujuan untuk mengurangi jumlah fading. Gambaran sederhana mengenai model komunikasi kooperatif ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Model Sistem Komunikasi Kooperatif

$h_{s,r}$: redaman kanal *source-relay*

$h_{s,d}$: redaman kanal *source-destination*

$h_{r,d}$: redaman kanal *relay-destination*

Source mengirimkan informasi ke *destination* dan *relay* dalam waktu yang sama. Sinyal yang diterima pada $y_{s,d}$, $y_{s,r}$ dapat dituliskan sebagai berikut :

$$y_{s,r} = \sqrt{P} h_{s,r}x + n_{s,r} \quad (2.1)$$

$$y_{s,d} = \sqrt{P} h_{s,d}x + n_{s,d} \quad (2.2)$$

Relay akan meneruskan informasi dari *source* ke *destination*. Sinyal yang diterima pada $y_{r,d}$ dapat dituliskan sebagai berikut :

$$y_{r,d} = h_{r,d}q(y_{s,r}) + n_{r,d} \quad (2.3)$$

Dimana :

$y_{s,r}$: Sinyal yang diterima pada *source-relay*

$y_{s,d}$: Sinyal yang diterima pada *source-destination*

$y_{r,d}$: Sinyal yang diterima pada *relay-destination*

P : Daya yang dikirim oleh sumber

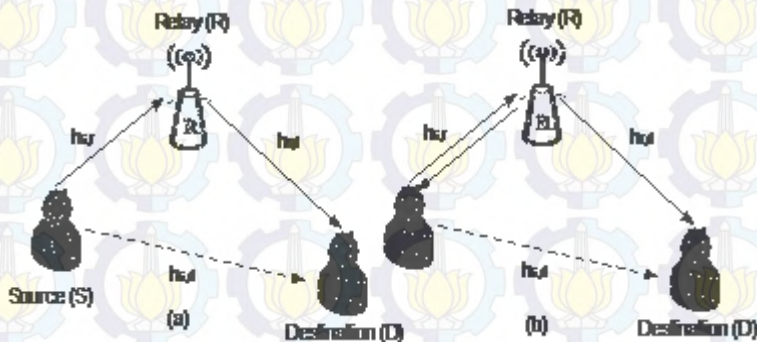
x : Simbol informasi yang dikirim

$n_{s,r}$, $n_{s,d}$, $n_{r,d}$: *Noise additive*

q : Fungsi $q(\cdot)$ tabel tergantung implementasi di *relay* node

2.3 Skenario Sistem Komunikasi Kooperatif

Pada sistem komunikasi kooperatif ada dua skenario yang digunakan untuk proses transmisi data, yang pertama skenario *relay* dan skenario simetri.



Gambar 2. 2 Skenario Komunikasi Kooperatif
(a) Skenario *relay* ; (b) Skenario simetri

Gambar 2.2 (a) menunjukkan transmisi data pada skenario *relay*. Pada skenario *relay*, *relay* merupakan terminal yang hanya digunakan untuk mengirim ulang data yang diterima menuju ke *destination*. Pada Gambar 2.2 (a), *source* (S) dapat mengirim data ke *destination* (D) melalui dua jalur, yaitu jalur langsung menuju D dan jalur melalui *relay* (R). Jadi R ini bukanlah sebuah *source*. Sedangkan pada skenario simetri, yang berfungsi sebagai *relay* adalah *user* lain yang berada di sekitarnya. Dalam proses transmisi, *user* lain harus menyediakan *slot* khusus untuk menampung data *user* lain. Gambar 2.2 (b) menunjukkan transmisi data pada skenario simetri. Pada Gambar 2.2 (b), S dan R adalah *source* sekaligus *relay*. S dapat mengirim data melalui dua jalur, yaitu jalur langsung menuju D dan jalur melalui R. Konsep tersebut berlaku juga pada R. R dapat mengirim data melalui dua jalur, yaitu jalur langsung menuju D dan jalur melalui S. Jadi pada skenario simetri setiap *user* memiliki dua fungsi, yaitu sebagai *source* dan sebagai *relay*. Skenario yang akan digunakan pada penelitian ini adalah skenario *relay*.

2.4 Protokol Relay pada Sistem Kooperatif

Relay merupakan terminal yang berfungsi untuk mengirimkan kembali data informasi yang telah diterima menuju ke penerima. Dimana

pengguna lain yang berada disekitarnya yang berfungsi sebagai relay. Pengolahan sistem kooperatif pada relay berbeda, sesuai dengan protokol yang bekerja. Teknik relay yang paling umum digunakan adalah protokol relay *Amplify and forward* (AF) dan protokol relay *Decode and forward* (DF) [3].

2.4.1. Protokol Relay *Amplify And Forward*

Pada protokol relay *amplify and forward*, sinyal yang diterima oleh relay yang terdapat noise dikuatkan terlebih dahulu untuk menormalisasi sinyal yang disebabkan oleh fading. Kemudian dikirimkan kembali ke tujuan tanpa melalui proses decoding. Ilustrasi *amplify and forward* [3] ditunjukkan pada gambar 2.3.

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 dengan menggunakan persamaan 2.4

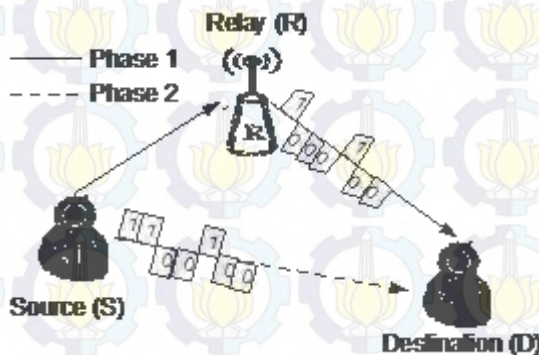
$$\beta \leq \sqrt{\frac{P}{|h_{s,r}|^2 P + N_0}} \quad (2.4)$$

keterangan :

P : power dari sinyal yang dikirimkan oleh pengirim [Watt]

$h_{s,r}$: koefisien fading antara *source* dan *relay*

N_0 : varian *noise*



Gambar 2. 3 *Amplify-and-Forward*

Skema relay *basic AF* meneruskan sinyal dari sumber ke tujuan tanpa memperhatikan kualitas link *s-r*. Pada fase I, sumber mengirim blok simbol $x_s = [x_s[0], \dots, x_s[M-1]]$ ke relay dan tujuan, dengan sinyal yang diterima dituliskan dengan persamaan:

$$y_r[m] = h_{s,r}\sqrt{P_s}x_s[m] + w_r[m] \quad (2.5)$$

$$y_d^{(1)}[m] = h_{s,d}\sqrt{P_s}x_s[m] + w_d^{(1)}[m] \quad (2.6)$$

dengan $m = 0, \dots, M-1$.

Pada fase II, relay menerima sinyal untuk menghasilkan vektor transmisi dinormalisasi x_r dengan $E[|x_r[m]|^2] = 1$ untuk semua m . Jika gain kanal saat itu $|h_{s,r}|^2$ diketahui pada relay, relay dapat mengalikan sinyal terima $y_r[m]$ dengan gain dituliskan dengan persamaan:

$$G_v = \frac{1}{\sqrt{E[|y_r[m]|^2/|h_{s,r}|^2]}} = \frac{1}{\sqrt{P_s|h_{s,r}|^2 + \sigma_r^2}} \quad (2.7)$$

untuk mendapatkan simbol transmisi di relay, dituliskan dengan persamaan:

$$x_r[m] = G_v y_r[m] \quad (2.8)$$

$$x_r[m] = \frac{P_s}{\sqrt{P_s|h_{s,r}|^2 + \sigma_r^2}} h_{s,r} x_s[m] + \frac{1}{\sqrt{P_s|h_{s,r}|^2 + \sigma_r^2}} w_r[m] \quad (2.9)$$

Catatan bahwa gain G_v tergantung pada kanal $s-r$ dan bervariasi pada interval transmisi yang berbeda. Jadi, skema ini sering dihubungkan sebagai skema relay AF variabel-gain. Kemudian, dengan daya P_r , relay meneruskan sinyal x_r ke tujuan, dengan sinyal diterima dengan persamaan:

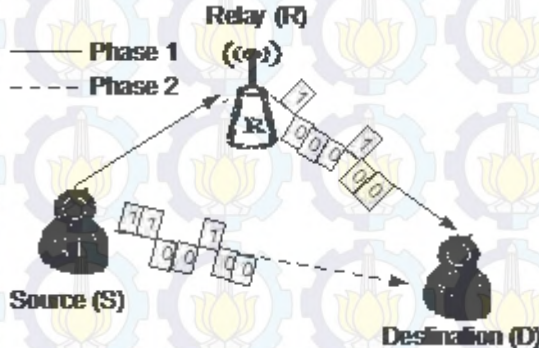
$$y_d^{(2)} = h_{r,d}\sqrt{P_r}x_r[m] + w_d^{(2)}[m] \quad (2.10)$$

$$y_d^{(2)} = \sqrt{\frac{P_s P_r}{P_s|h_{s,r}|^2 + \sigma_r^2}} h_{s,r} h_{r,d} x_s[m] + \sqrt{\frac{P_s}{P_s|h_{s,r}|^2 + \sigma_r^2}} h_{r,d} w_r[m] + w_d^{(2)}[m] \quad (2.11)$$

dengan $m = 0, \dots, M - 1$. Sinyal tiba di tujuan dapat digunakan untuk deteksi dengan atau tanpa *diversity combining* [8].

2.4.2. Protokol Relay *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 dikirm tersebut. Pada metode *decode and forward*, sinyal informasi yang diterima *direlay* 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 2.4.



Gambar 2. 4 *Decode-and-Forward*

2.5 Teknik *Combining* [8]

Teknik *combining* yang biasa dipakai untuk ketiga jenis teknik *diversity* adalah *selective combining*, *equal gain combining*, dan *maximum ratio combining*.

a. Selective combining (SC)

SC merupakan teknik *combining* yang paling sederhana dan sering digunakan. Dengan menggunakan beberapa cabang antena di mana setiap cabang antena 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.

$$Z_{SC}[n] = \max(y_k[n]) \quad (2.12)$$

keterangan :

$Z_{SC}[n]$: sinyal hasil SC di *destination*
 $y_k[n]$: sinyal yang diterima *destination*
 k : 1,2,3,... n

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.

a. Equal gain combining (EGC)

Teknik ini merupakan teknik *combining* yang menjumlahkan semua sinyal informasi yang diterima. Setelah semua sinyal yang diterima dijumlahkan, selanjutnya hasil penjumlahan dari sinyal tersebut akan dideteksi untuk mendapatkan estimasi data yang dikirimkan. Hasil dari *equal gain combining* dituliskan pada persamaan 2.13 :

$$Z_{EGC}[n] = \sum_{i=1}^k \alpha_k y_k[n] \quad (2.13)$$
$$\alpha_k = e^{-j\phi_k}$$

keterangan :

$Z_{EGC}[n]$: sinyal hasil EGC di *destination*
 $y_k[n]$: sinyal yang diterima *destination*
 k : jumlah link
 α_k : faktor pengali

Teknik ini lebih baik daripada teknik *selective combining* karena pada teknik ini semua sinyal yang diterima diproses secara simultan untuk mengestimasi data yang dikirimkan.

b. Maximum ratio combining (MRC)

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. Hasil dari *maximum ratio combining* dituliskan pada persamaan 2.14 .

$$Z_{MRC}[n] = \sum_{i=1}^k \alpha_k y_k[n] \quad (2.14)$$

$$\alpha_k = |h_k| e^{-j\phi_k} / \sigma_k^2$$

keterangan :

$Z_{EGC}[n]$: sinyal hasil EGC di *destination*

$y_k[n]$: sinyal yang diterima *destination*

k : jumlah link

$|h_k|$: respon kanal

σ_k^2 : *noise variance*

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 Modulasi QPSK

Quadrature Phase Shift Keying (QPSK) memiliki dua kali efisiensi *bandwidth* jika dibandingkan dengan *Binary Phase Shift Keying* (BPSK), di mana 2 bit ditransmisikan dalam satu simbol modulasi. Fase pembawa membawa 1 dari 4 spasi nilai yang sama, seperti $0, \pi/2, \pi,$ dan $3\pi/2,$ di mana masing-masing nilai dari fase tersebut merepresentasikan pasangan bit yang unik. Sinyal modulasi dapat ditulis sebagai [1]

$$S_i(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f_c t + \frac{2\pi}{M}(i-1)\right), 0 \leq t \leq T_s \quad (2.15)$$

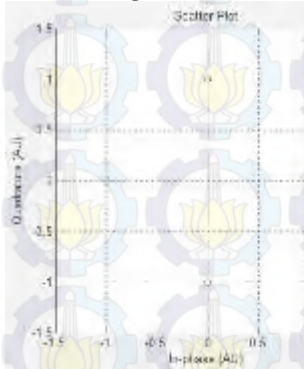
dimana $E_s = (\log_2 M)E_b$ adalah energi per simbol dan $T_s = (\log_2 M)T_b$ adalah simbol periode. Dari persamaan tersebut, dapat kita tuliskan ke dalam bentuk *quadrature*

$$S_i(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left[i-1 \frac{2\pi}{M}\right] \cos(2\pi f_c t) - \sqrt{\frac{2E_s}{T_s}} \sin\left[i-1 \frac{2\pi}{M}\right] \sin(2\pi f_c t) \quad (2.16)$$

QPSK merupakan teknik pengkodean M -ary PSK dimana $M = 4$. Dalam modulasi QPSK terdapat empat *phase output* yang berbeda, maka

harus ada empat kondisi input yang berbeda. Karena input digital ke modulator QPSK adalah sinyal biner, maka untuk menghasilkan 4 kondisi input yang berbeda harus dipakai bit input lebih dari 1 bit tunggal. Dalam hal ini, yaitu menggunakan 2 bit, sehingga terdapat empat kondisi yang mungkin yaitu 00, 01, 10, dan 11 [9].

Variasi dari empat fasa QPSK, atau dapat disebut $\frac{\pi}{4}$ -QPSK, dapat diperlihatkan konstelasinya pada Gambar 2.6. *Phase offset* yang digunakan pada konstelasi tersebut adalah 0° . Apabila diterapkan *phase offset* $\pi/4$ menyebabkan konstelasi QPSK berputar 45° atau serupa dengan konstelasi *Quadrature Amplitude Modulation* (QAM).



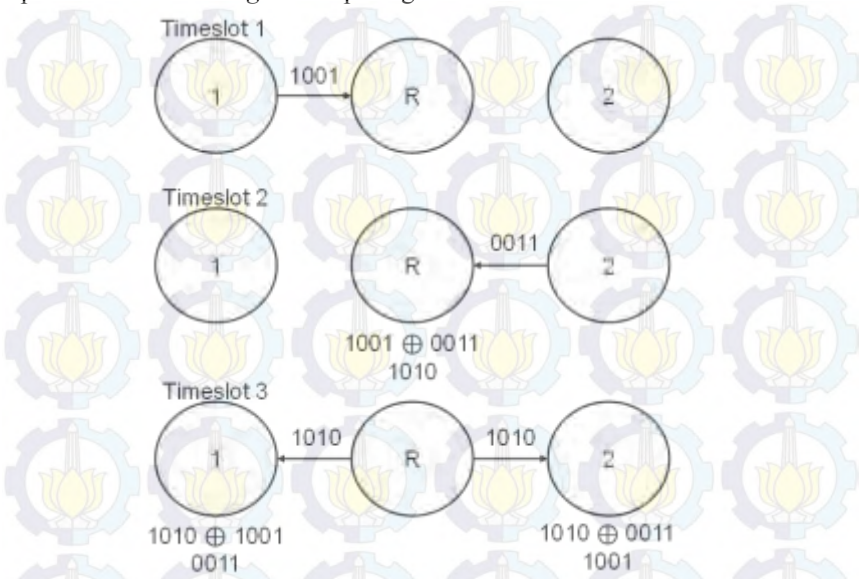
Gambar 2. 5 Konstelasi QPSK pada Matlab

2.7 Network Coding

Konsep *Network Coding* pertama kali diperkenalkan pada tahun 2000 oleh Ahlswede, Cai, Li, dan Yeung. Kinerja *network coding* dibandingkan dengan *non network coding* lebih baik dan efisien karena data yang dikirim dikombinasikan dengan pengkodean dari beberapa source dengan satu pengiriman data. Dengan menggunakan *network coding* diharapkan masalah kecepatan download file, efisiensi dalam distribusi dapat diselesaikan, sehingga pengguna jaringan lebih mudah dalam melakukan akses download file dalam jumlah besar [4]. *Network coding* dapat meningkatkan *throughput*, ketahanan, kompleksitas, dan keamanan. Potensi keuntungan *network coding* adalah efisiensi sumber daya (*bandwidth* dan *power*) dan efisiensi komputasi.

Terdapat 3 konsep *network coding* yaitu skema *tradisional* tanpa *network coded* (TS), skema *straightforward network coding* (SNC) dan

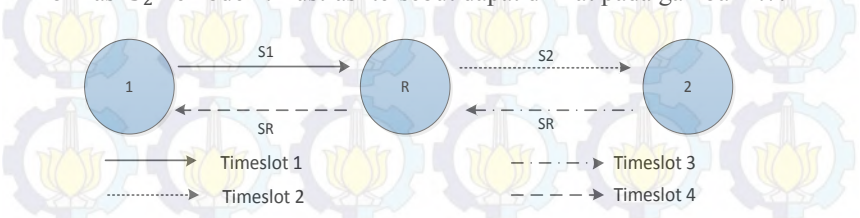
skema *physical layer network coding* (PNC) [5]. Skema pengiriman proses *network coding* adalah pada gambar berikut:



Gambar 2. 6 Skema pengiriman pada *network coding*

2.7.1 Skema Tradisional Tanpa Network Coding (TS)

Skema Pada skema tradisional tanpa *network coding*, dengan menggunakan 4 timeslot pada menggunakan 2 *packets*. Pada timeslot 1, node 1 mengirimkan paket S_1 ke *relay* R. Pada timeslot 2, *Relay* R mengirimkan kembali S_1 ke node 2. Pada timeslot 3, node 2 mengirim paket S_2 ke *relay* R dan Pada timeslot 4, *relay* R mengirimkan kembali informasi S_2 ke node 1. Ilustrasi tersebut dapat dilihat pada gambar 2.7.



Gambar 2. 7 Tradisional Tanpa *Network Coding*

2.7.2 Skema Straightforward Network Coding (SNC)

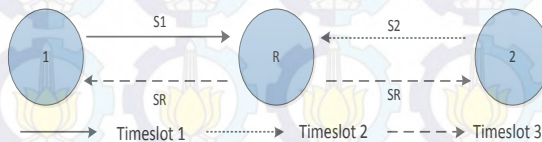
Pada *straightforward network coding*, menggunakan timeslot 3 sehingga dapat memiliki kemampuan untuk menaikkan throughput sebesar 33%. Pada gambar 2.8 menggambarkan ilustrasi SNC menggunakan 3 timeslot untuk proses pengiriman data informasi. Pada timeslot 1, node 1 mengirimkan S_1 ke *relay*. Kemudian pada timeslot 2, node 2 mengirimkan informasi S_2 ke *relay*. Setelah menerima informasi dari S_1 dan S_2 , *relay* kemudian membantu 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 berikut :

$$S_R = S_1 \oplus S_2 \quad (2.17)$$

Pada timeslot 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 :

$$S_1 \oplus S_2 = S_1 \oplus (S_1 \oplus S_2) = S_2 \quad (2.18)$$

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 timeslot yang berbeda.

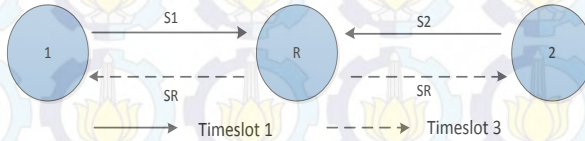


Gambar 2. 8 Skema *Straightforward Network Coding*

2.7.3 Skema Physical Layer Network Coding (PNC)

Pada *physical layer network coding* diharapkan dapat mengurangi jumlah time slot, dengan hanya membutuhkan dua timeslot 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 2.9 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 $S_R = S_1 \oplus S_2$. Kemudian pada timeslot kedua, relay R membroadcast S_R ke node 1 dan node 2.



Gambar 2. 9 Skema Physical Layer Network Coding

Data informasi dari node 1 dan node 2 diterima di relay dengan fase dan amplitudo yang sama. Pada pemetaan PNC, gelombang EM ditambahkan dengan *noise* untuk paket *output* yang diteruskan oleh *relay*. Kunci utama dalam PNC adalah bagaimana *relay* membentuk $S_R = S_1 \oplus S_2$ dari gelombang elektromagnetik. Proses ini disebut dengan “PNC *mapping*”. Secara umum, *mapping* PNC adalah proses pemetaan dari gelombang EM yang diterima ditambah *noise* menjadi beberapa paket untuk diteruskan oleh *relay*. Semua *mapping* PNC membagi kunci yang dibutuhkan oleh node 1 dan node 2 untuk menghasilkan informasi dari node lain berdasarkan paket yang dihasilkan oleh *relay* dan informasi node itu sendiri.

Semua node menggunakan modulasi QPSK untuk mengirimkan sinyal, level simbol dan fasa carrier dianggap saling sinkron. Untuk sementara *noise* diabaikan. Pada satu bagian periode simbol, node 1 dan 2 memodulasi simbol dengan RF pada frekuensi ω , maka dituliskan node i mengirimkan sinyal $Re[(a_i + jb_i)e^{j\omega t}]$. Sinyal kombinasi bandpass yang diterima oleh R *relay* selama satu periode adalah :

$$\begin{aligned} y_R(t) &= S_1(t) + S_2(t) \\ &= [a_1 \cos(\omega t) - b_1 \sin(\omega t)] + [a_2 \cos(\omega t) - b_2 \sin(\omega t)] \\ &= (a_1 + a_2) \cos(\omega t) - (b_1 + b_2) \sin(\omega t) \end{aligned} \quad (2.19)$$

Dimana :

$S_i(t), i \in \{1,2\}$: Sinyal bandpass yang dikirimkan oleh node i
 $a_i \in \{-1,1\}$ & $b_i \in \{-1,1\}$: Hasil bit dari modulasi QPSK.

Untuk QPSK, $a_i=1$ dihubungkan dengan bit 0 dan $a_i=-1$ dihubungkan dengan bit 1 dalam sinyal inphase. Begitu juga dengan b_i

dalam sinyal quadrature phase. Sehingga XOR dapat di tulis menjadi perkalian :

$$a_1 \oplus a_2 = a_1 \cdot a_2$$

$$b_1 \oplus b_2 = b_1 \cdot b_2$$

Komponen inphase (I) dan quadrature (Q) ditulis dengan persamaan:

$$y_R^{(I)} = a_1 + a_2 \quad (2.22)$$

$$y_R^{(Q)} = b_1 + b_2 \quad (2.23)$$

Relay R tidak bisa menemukan informasi simbol yang dikirim oleh node 1 dan 2 dalam bentuk (2.22 dan 2.23) karena $y_R^{(I)}$ dan $y_R^{(Q)}$ memberikan dua persamaan dengan empat variable yang tidak diketahui: a_1, b_1, a_2, b_2 .

Dalam PNC, Relay R tidak membutuhkan nilai-nilai dari empat variable itu tetapi hanya membutuhkan untuk mengirimkan $a_1 \oplus a_2$ dan $b_1 \oplus b_2$ untuk menghasilkan PNC mapping $S_R \cong a_1 \oplus a_2 + j(b_1 \oplus b_2) \cong a_R + j b_R$. Sebagian persamaan $a_1 \oplus a_2$ dan $b_1 \oplus b_2$ bisa didapat dari $y_R^{(I)}$ dan $y_R^{(Q)}$. Maka bisa ditulis fungsi dari PNC mapping $S_R = y_R^{(I)} + j y_R^{(Q)}$. PNC mapping dapat ditulis sebagai berikut :

$$a_R = \begin{cases} -1 & \text{jika } y_R^{(I)} = 0 \\ 1 & \text{jika } y_R^{(I)} = -2 \text{ atau } 2 \end{cases} \quad (2.24)$$

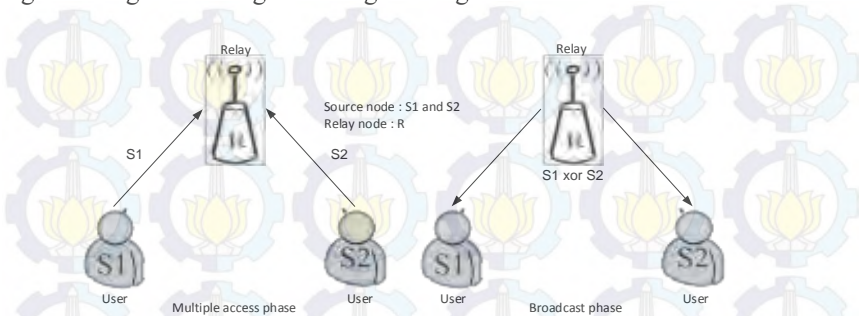
Setelah pemetaan PNC, relay R mengirim sinyal ke node 1 dan node 2 sebagai berikut :

$$S_R(t) = a_R \cos(\omega t) - b_R \sin(\omega t) \quad (2.25)$$

Dalam PNC, sinyal RF berada timeslot kedua tetapi pada konvensional network coding sinyal RF dikirimkan pada timeslot ketiga. Perbedaan yang utama dari dua skema ini adalah dari cara mengirimkan (a_R, b_R) . Dalam PNC (a_R, b_R) dikirimkan oleh node 1 dan 2, kemudian relay R secara khusus melakukan decoding (a_1, b_1) dan (a_2, b_2) untuk dijadikan bentuk (a_R, b_R) . Perhatikan bahwa $(a_1 + a_2, b_1 + b_2)$ adalah bentuk dari operasi network coding.

Data informasi dari node 1 dan node 2 diterima di relay dengan fase dan amplitudo yang sama. Pada pemetaan PNC, gelombang EM ditambahkan dengan *noise* untuk paket *output* yang diteruskan oleh *relay*.

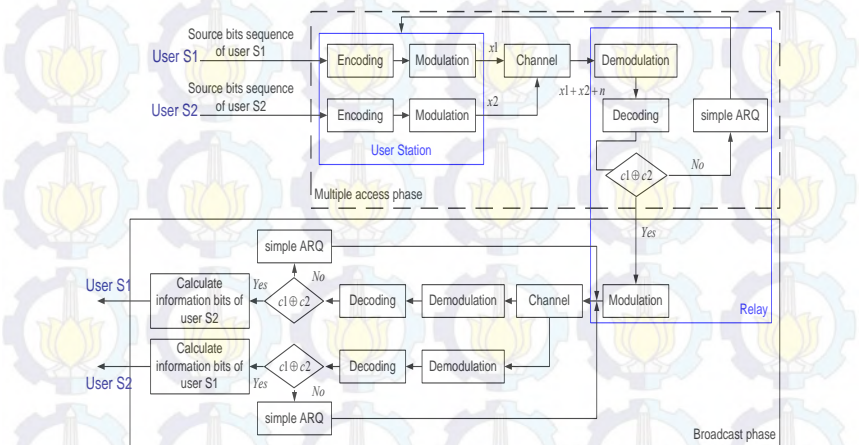
Pada PNC, proses deduksi terjadi pada relay $S_R = S_1 \oplus S_2$ dari gelombang elektromagnetik dengan mengabaikan *noise*.



Gambar 2. 10 System model of PNC [10]

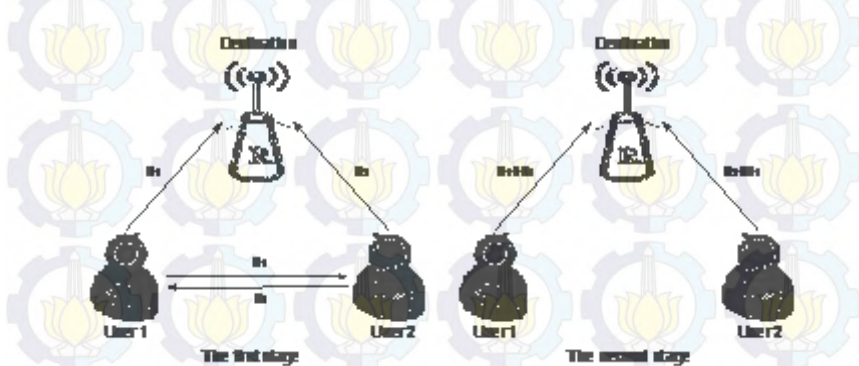
Catatan:

- Multiple access phase : Kedua User (S1 dan S2) akan mengirimkan informasi secara bersamaan ke Node Relay dalam 1 timeslot
- Broadcast phase : Pada node Relay informasi awal (S1 dan S2) akan di *decode* dan dikirimkan kembali ke User (S1 dan S2) dengan menggunakan teknik XOR.



Gambar 2. 11 Blok Diagram skema PNC

Pada PNC juga terdapat skema AF (*amplify and forward*) yang disebut dengan (*Pseudo Physical-Layer Network Coding amplify and forward (PPNCAF)*) [7], seperti yang diilustrasikan pada gambar 2.12.



Gambar 2. 12 System Model of PPNCAF

Seperti terlihat pada gambar 2.12, pada sistem terdapat 2 terminal yaitu node *user* dan node *destination* (BS). Untuk skema PPNCAF *timeslot* 1, *user* 1 dan *user* 2 mentransmisikan dua buah pesan yaitu s_1 dan s_2 secara berturut-turut menggunakan *channel coding*. Kedua pesan saling menggunakan kanal orthogonal. pada *timeslot* ke 2, pesan tersebut di *encode* menjadi s_1+s_2 dan s_2+s_1 menggunakan teknik *pseudo physical-layer network coding* pada node *relay* secara berturut-turut. Pada node *destination* menerima 4 jenis pesan yang berbeda yaitu s_1 , s_2 , s_1+s_2 , dan s_2+s_1 , masing-masing dari empat pesan ini mengalami fading. Salah satu dari dua pesan dalam *timeslot* yang sama dengan satu pesan di *timeslot* yang lain dapat membangun kembali dua pesan sumber yaitu s_1 dan s_2 . Node *destination* merekonstruksi pesan tersebut dengan melakukan MRC (*Maximum ratio combining*)

2.7.4 Analisis Performansi *Pseudo Physical Layer Network Coding amplify and forward (PPNCAF)* [7]

Skema PPNCAF bisa dibagi menjadi dua tahapan, yaitu pada *timeslot* 1 dan *timeslot* 2. Pesan yang dikirimkan dari setiap node dapat dituliskan dengan persamaan:

$$y_{i,j,k} = h_{i,j,k}x_{i,j,k} + n_{i,j,k} \quad (2.26)$$

Keterangan : $x_{i,j,k}$ dan $y_{i,j,k}$ adalah pesan yang dikirim dan diterima oleh node i secara berturut-turut. $n_{i,j,k}$ additif white Gaussian noise; dan $i = 1,2$ menunjukkan *user* transmisi 1 dan transmisi 2; $j, j \in (1, 2, d)$ menunjukkan penerimaan oleh *user* 1, 2 dan BS secara berturut-turut, dan k menunjukkan *timeslot*.

Pada tahap pertama, *user* 1 dan *user* 2 membroadcast pesan satu dengan yang lain dan juga ke node *destination* secara berturut-turut tanpa memperhatikan kualitas link *s-r*. Pesan yang diterima oleh kedua node ini ditulis dengan persamaan:

$$\begin{aligned} y_{1,2,1} &= \sqrt{P_2} h_{1,2,1} x_{2,1,1} + n_{1,2,1} \\ y_{2,1,1} &= \sqrt{P_1} h_{2,1,1} x_{1,2,1} + n_{2,1,1} \end{aligned} \quad (2.27)$$

Dimana $h_{1,2,1} = h_{2,1,1}$. *User* 1 dan 2 menyimpan pesan yang telah diterima di node masing-masing yang telah saling dikirimkan satu sama lain. Sementara itu, pesan yang diterima pada BS ditulis dengan persamaan:

$$\begin{aligned} y_{d,1,1} &= \sqrt{P_1} h_{d,1,1} x_{1,d,1} + n_{d,1,1} \\ y_{d,2,1} &= \sqrt{P_2} h_{d,2,1} x_{2,d,1} + n_{d,2,1} \end{aligned} \quad (2.28)$$

Dimana P adalah power dari sinyal yang diterima (Watt), $x_{1,1,1} = x_{1,d,1} = s_1$, $x_{2,1,1} = x_{2,d,1} = s_2$, dan $h_{d,1,1} = h_{d,2,1}$. Pada tahap kedua di *user* 1 dan 2 menggabungkan pesan masing-masing dengan pesan yang lainnya dengan mengkalikan faktor β dengan teknik *pseudo physical-layer network coding* (Sebuah skema PNC yang memungkinkan penyimpanan pada *relay* dan menggabungkan pesan tersebut tanpa mempertimbangkan pengaruh *asynchronous*). Kombinasi menggabungkan pesan pada *timeslot* kedua ini ditulis dengan persamaan:

$$\begin{aligned} x_{1,d,2} &= \beta y_{1,2,1} + s_1 \\ x_{2,d,2} &= \beta y_{2,1,1} + s_2 \end{aligned} \quad (2.29)$$

Dan pesan yang diterima pada node *destination* (BS) pada *timeslot* kedua ini adalah:

$$\begin{aligned} y_{d,1,2} &= \sqrt{P_1} h_{d,1,2} x_{1,d,2} + n_{d,1,2} \\ y_{d,2,2} &= \sqrt{P_2} h_{d,2,2} x_{2,d,2} + n_{d,2,2} \end{aligned} \quad (2.30)$$

Menggunakan persamaan 2.27 dan 2.29 di node *destination* (BS) pesan ini akan diproses menjadi:

$$\begin{aligned} z_1 &= y_{d,2,2} - y_{d,2,1} \\ z_2 &= y_{d,1,2} - y_{d,1,1} \end{aligned} \quad (2.31)$$

Dengan mempertimbangkan pesan yang diransmisikan oleh *user* 1, pemrosesan pesan bisa mengikuti metode *Maximal-Ratio Combining* (MRC), sehingga didapatkan:

$$\hat{y}_1 = \frac{\sqrt{P_1}h_{d,1,1}^*}{N_{d,1,1}} y_{d,1,1} + \frac{\beta\sqrt{P_1P_2}h_{2,1,1}^*h_{d,2,2}^*}{\beta^2h_{d,2,2}^2P_2N_{2,1,1} + N_{d,2,2} - N_{d,2,1}} z_1 \quad (2.32)$$

Karena *user* 1 dikombinasikan sendiri dengan s1 dengan pesan yang diterima $y_{1,2,1}$ pada tahap kedua, dimana pesan $y_{1,2,1}$ diterima dari *user* 2 pada tahap pertama. Dengan memproses pesan $y_{d,1,2}$ dan $y_{d,2,1}$ didapatkan:

$$\begin{aligned} z_3 &= y_{d,1,2} - y_{d,2,1} \\ &= \sqrt{P_1}h_{d,1,2}(\sqrt{P_2}h_{1,2,1}s_2 + n_{1,2,1} + s_1) + n_{d,1,2} - (\sqrt{P_2}h_{d,2,1}s_2 + n_{d,2,1}) \\ &= \sqrt{P_1}h_{d,1,2}s_1 + n_{\Delta} \end{aligned} \quad (2.33)$$

Dimana: $n_{\Delta} = (\sqrt{P_1P_2}h_{d,1,2}h_{1,2,1} - \sqrt{P_2}h_{d,2,1})s_2 + \sqrt{P_1}h_{d,1,2}n_{1,2,1} + n_{d,1,2} - n_{d,2,1}$

Shingga pesan pada s1 dikirimkan oleh *user* 1 akan sama dengan *retransmission mode*, sehingga akan mendapatkan pesan s2 yang sama persis. Selain itu, ketika user i , $i \in (1, 2)$ tidak memiliki pesan yang akan dikirimkan, maka teknik dengan menggunakan skema PPNCAF bisa dirubah menjadi skema sistem kooperatif AF.

Untuk menganalisis *outage probability* pada sistem kooperatif AF dengan PNC ini dengan menggunakan kanal Gaussian dengan level *noise* berbeda-beda pada outputnya. Nilai maximum rata-rata informasi yang sama antara input dan dua buat output pada kanal input Gaussian ditulis:

$$I_{PPNCAF} = \frac{1}{2} \log \left(1 + \frac{P_1}{N_0} |h_{d,1,1}|^2 + f \left(\frac{P_1}{N_0} |h_{2,1,1}|^2, \frac{P_2}{N_0} |h_{d,2,2}|^2 \right) \right) \quad (2.34)$$

Asumsikan kanal yang digunakan adalah Rayleigh kanal, sehingga nilai *outage probability* dapat dicari dengan persamaan:

$$P_1 = (I_{PPNCAF} < R) \\ = Pr \left(|h_{d,1,1}|^2 + \frac{1}{SNR} f \left(SNR|h_{2,1,1}|^2, SNR|h_{d,2,2}|^2 \right) < g \right) \quad (2.35)$$

Ketika $g = \frac{2^{2R}-1}{SNR}$ nilai SNR dapat diperkirakan ketika kondisi *medium to high* :

$$P_1 = \left(\frac{1}{2E \left\{ |h_{d,1,1}|^2 \right\}} \frac{E \left\{ |h_{2,1,1}|^2 \right\} + E \left\{ |h_{d,2,2}|^2 \right\}}{E \left\{ |h_{2,1,1}|^2 \right\} E \left\{ |h_{d,2,2}|^2 \right\}} \right) \left(\frac{2^{2R} - 1}{SNR} \right)^2 \quad (2.36)$$

Berdasarkan persamaan 2.36, pesan yang disampaikan oleh *user 1* dan *user 2* dengan Skema PPNCAF dapat dicapai kondisi *diversity order*.

Jika kondisi kanal *user j*, $j \in (1,2)$, $j \neq i$ untuk node *destination* pada tahap kedua lebih buruk, maka selama kondisi kanal dari *user i* ke node *destination* belum penuh, maka node *destination* bisa memproses pesan si kedalam urutan yang seragam. Dengan demikian *outage probability* dari skema PPNCAF dapat ditulis dengan persamaan:

$$P_{out,PPNCAF} = MIN(Pr_1, Pr_2) \quad (2.37)$$

Dimana:

$$Pr_1 = Pr \left(|h_{d,1,1}|^2 + \frac{1}{SNR} f \left(SNR|h_{2,1,1}|^2, SNR|h_{d,2,2}|^2 \right) < \frac{2^{2R} - 1}{SNR} \right), \\ Pr_2 = Pr \left(|h_{d,2,1}|^2 + \frac{1}{SNR} f \left(SNR|h_{1,2,1}|^2, SNR|h_{d,1,2}|^2 \right) < \frac{2^{2R} - 1}{SNR} \right)$$

2.8 Software Defined Radio

Software Defined Radio (SDR) adalah teknologi yang memungkinkan untuk mengubah parameter RF, frekuensi pembawa, jenis modulasi, atau daya output sebuah pemancar atau penerima radio menggunakan sebuah *software*. *Software Defined Radio* (SDR) menggunakan perangkat digital yang dapat diprogram untuk melakukan pemrosesan sinyal yang diperlukan untuk mengirim dan menerima informasi *baseband* pada frekuensi radio. Perangkat digital seperti *signal source processor* (DSP) dan *field programmable gate array* (FPGA)

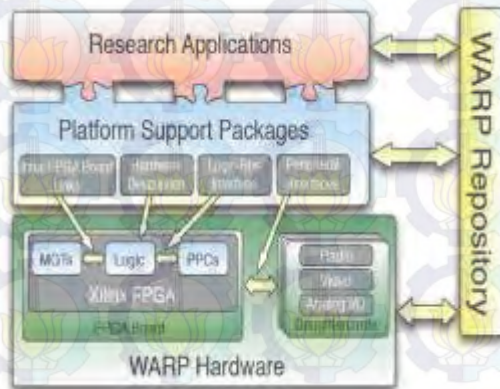
membutuhkan sebuah fungsi untuk pemrosesan sinyal. SDR dapat menyesuaikan dengan *software* terbaru jika muncul standar komunikasi nirkabel yang baru.

2.9 *Wireless Open-Access Research Platform* dan WARPLab [6]

Wireless Open-Access Research Platform (WARP) adalah perangkat radio yang bisa diprogram untuk membuat *prototype* jaringan *wireless*. WARP merupakan salah satu bentuk *software* SDR sebagai modul penelitian tentang sistem komunikasi nirkabel. WARP ini juga merupakan sebuah *Field Programmable Gate Array* (FPGA) atau modul *Digital Signal Processing* (DSP) yang terhubung dengan modul radio.

Pengembangan platform WARP ini memiliki banyak keuntungan yaitu, dapat diimplementasikan untuk algoritma *wireless* yang lebih kompleks secara *real time*, dapat dikonfigurasi terutama di desain untuk *prototype* algoritma komunikasi nirkabel untuk aplikasi yang berorientasi pada edukasi dan *research*.

Arsitektur dari *platform* WARP didesain untuk memenuhi permintaan dari penelitian *high-performance* sistem *wireless*. Arsitektur platform WARP dengan 4 komponen utama ditunjukkan pada gambar 2.13.



Gambar 2. 13 Komponen-komponen utama dari arsitektur WARP

1. *Custom Hardware*: desain *Custom hardware* disesuaikan untuk kebutuhan komunikasi *wireless* dengan kinerja yang tinggi. *Hardware* menyediakan sumber daya tinggi yang mampu

diimplementasikan pada kecepatan algoritma, dapat terukur interkoneksinya untuk mengalokasikan lebih banyak kekuatan pemrosesan yang diperlukan, dan pilihan periferil yang diperluas untuk radio, prosesor khusus dan antarmuka ekspansi lainnya.

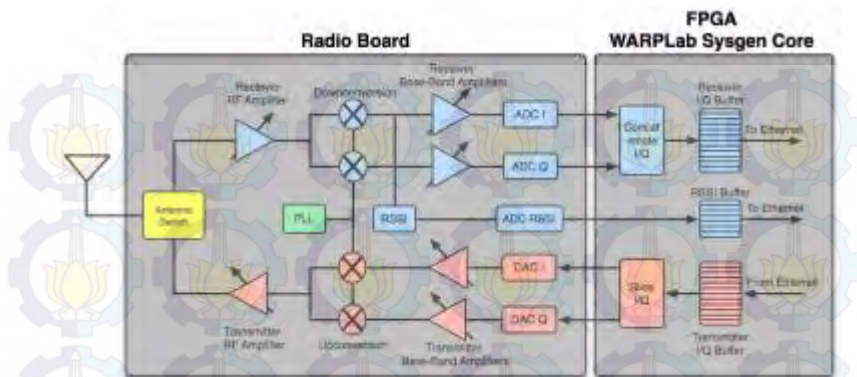
2. *Platform Support Packages*: perangkat-perangkat dan modul *interface* level rendah (*low-level*) yang didesain untuk memungkinkan penggunaan perangkat oleh para peneliti pada semua lapisan (*layer*) dari desain jaringan *wireless*.
3. *Open-Access Repository*: Penyimpanan untuk semua file-file *source-code*, model dan desain *hardware*.
4. *Research Application*: Implementasi dari algoritma-algoritma, dengan *custom hardware* dan paket pendukung platform.



Gambar 2. 14 Modul WARP Versi 2.2

2.9.1 WARP Board

Proses implementasi sistem komunikasi pada WARP dilakukan pada modul WARP sebagai FPGA dan modul radio sebagai interface antara sistem komunikasi dengan pengolahan sinyal digital. Secara umum, sistem WARP ditunjukkan dengan diagram blok pada gambar 2.14.



Gambar 2. 15 Diagram blok node WARP

Dalam proses pengiriman dan penerimaan sinyal informasi, pengolahan digital dilakukan dengan menggunakan FPGA dan dengan *radio board*. Dari gambar 2.14 menunjukkan jalur pemancar dan penerima, blok pemancar ditandai dengan warna merah dan blok penerima ditandai dengan warna biru.

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*: mengkonversi 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 sinyal 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.

Pada masing-masing radio board, akan terdapat dua *buffer* yaitu *In-phase* dan *Quadrature*. Proses pengiriman sinyal dari WARP Lab misalnya, akan menggunakan sinyal kompleks. Komponen utama dari node WARP yaitu :

a. FPGA board

FPGA pada WARP merupakan prosesor utama karena memberikan pengolahan sumber daya yang luar biasa, blok *programmable logic* didalamnya dapat saling dihubungkan sehingga dapat membentuk unit fungsional yang kompleks. FPGA board merupakan inti dari proses SDR.

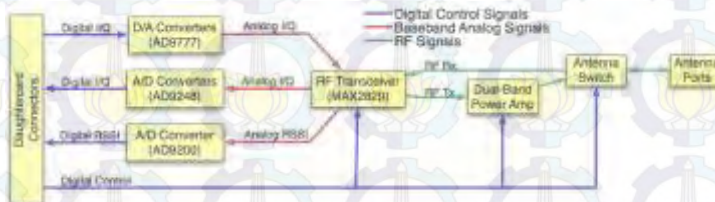
b. Radio board

Radio board merupakan interface dari domain digital ke domain analog. Pada proses konversi dari digital ke analog dan sebaliknya, *radio board* menggunakan A/D (*Analog-to-Digital*) dan D/A (*Digital-to-Analog*) *converter*. *Radio board* juga melakukan proses modulasi dan demodulasi *baseband*. *Radio board* pada WARP didesain menggunakan IC Maxim 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 -2.5GHz dan 4.9GHz dan

5.875 GHz. Bentuk fisik dari radio *board* ditunjukkan pada gambar 2.16 dan arsitektur dari radio board WARP ditunjukkan pada gambar 2.17.



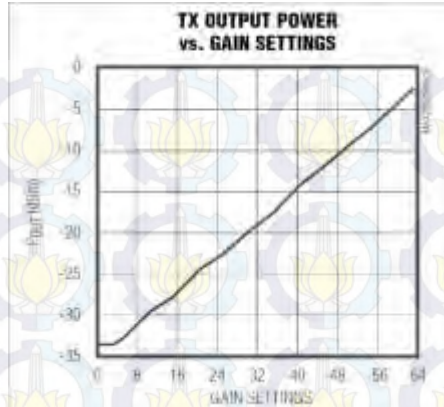
Gambar 2. 16 Radio board



Gambar 2. 17 Arsitektur radio board

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

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 [11]

Berdasarkan referensi dari data sheet MAX2829 [11] karakteristik RF gain memiliki kaitan sebagai berikut :

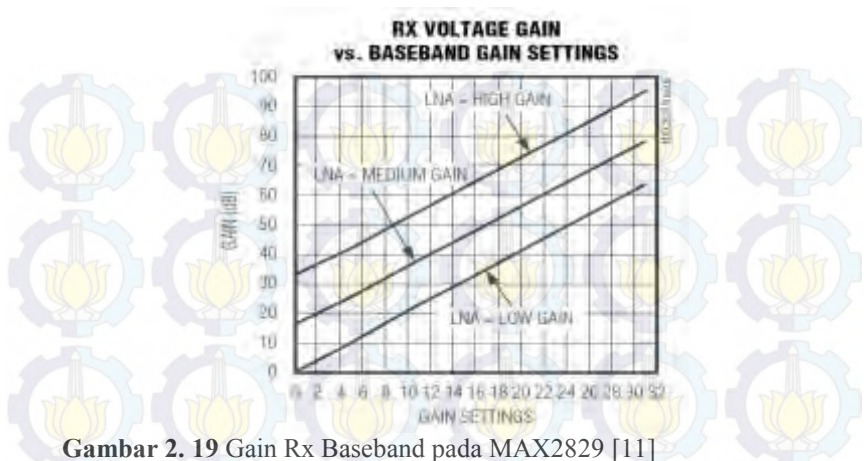
- Nilai gain linear pada range 4 hingga 63.
- Daya output dinyatakan dalam persamaan terhadap gain setting sebagai berikut :

$$P_{out} = 0.5252x - 36.1017$$

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 [11]

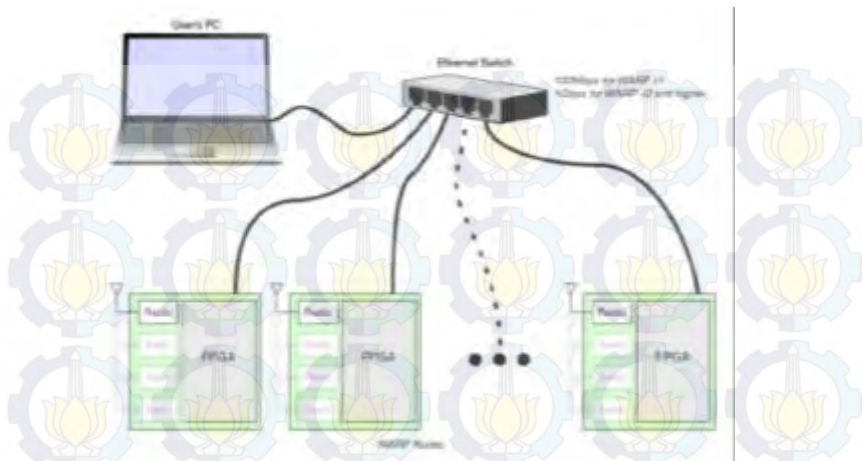
RSSI juga digunakan untuk mendapatkan nilai daya terima. Pembacaan RSSI menggunakan 10-bit linear dalam dB yang dapat dikonversi menjadi daya terima dalam dBm. RSSI diukur setelah RF amplifier, sehingga konversi nilai RSSI menjadi daya terima bergantung pada pengaturan Gain RX_RF [Maxim].

- 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.

2.9.2 WARPLab

WARPLab merupakan sebuah kerangka kerja yang merupakan digunakan untuk desain purwarupa *physical layer*. Penggunaan dari WARPLab memungkinkan kombinasi dari beberapa sistem multi-antena pada pemancar dan penerima. Kerangka kerja yang digunakan memiliki fleksibilitas untuk pengembangan sistem dengan banyak node. Logika dari kerangka kerja WARPLab ditunjukkan pada gambar 2.20.

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



Gambar 2. 20 Penggunaan Kerangka Kerja WARPLab [6]

BAB III

PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan membahas tentang proses perancangan dan implementasi sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* dan tradisional pada modul WARP. Untuk memudahkan proses perancangan dan implementasi sistem diperlukan alur yang menjelaskan garis besar proses yang dilakukan pada saat melaksanakan tugas akhir.

3.1 Alur Perancangan Sistem

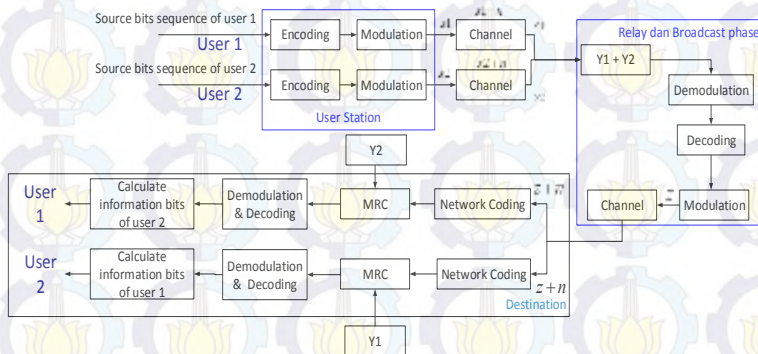
Dalam melakukan perancangan sistem perlu dilakukan beberapa tahapan. Langkah pertama desain sistem, lalu pengintegrasian PC dengan hardware WARP. Langkah kedua yaitu memodelkan sistem komunikasi kooperatif dengan skema *network coding*, kemudian menerapkan sistem komunikasi kooperatif tersebut dengan teknik modulasi QPSK menggunakan MATLAB untuk pengolahan data sebelum dikirim dan setelah diterima. Langkah berikutnya yaitu mengimplementasikan sistem yang dibuat secara *real time* menggunakan perangkat WARP. Dari pengimplementasian sistem akan diperoleh nilai BER untuk di analisis kinerjanya. Adapun alur perancangan sistem ditunjukkan pada gambar 3.1.



Gambar 3. 1 Alur Perancangan Sistem

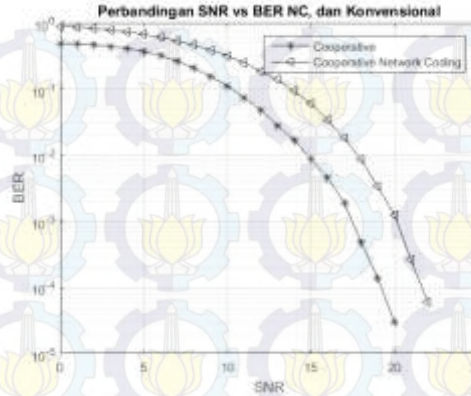
3.2 Desain Sistem

Sebelum dilakukan implementasi ke modul WARP terlebih dahulu mendesain sistem yang akan diimplementasikan. Desain sistem pada perencanaan tugas akhir ini yaitu melakukan simulasi pada MATLAB. Tahap pertama yaitu *user 1* dan *user 2* mengirimkan informasinya ke *node relay* dan ke *node user* lainnya yang sudah tercampur *noise* (kanal), setelah informasi tersebut di *encoding* dan di modulasi menggunakan modulasi QPSK. Lalu untuk di *node relay*, simbol informasi dari *user 1* dan *user 2* dilakukan proses *network coding* dengan penjumlahan simbol *user 1* dan *user 2* menggunakan konsep PNC (*physical network coding*) dan kemudian di beri penguatan (*amplify and forward*) lalu di *decoding* dan di modulasi QPSK setelah itu informasi tersebut di *broadcast* ke masing-masing *user 1* dan *user 2* (*destination*). Informasi yang di *broadcast* akan terjadi penambahan *noise*. Pada *destination* tersebut dilakukan proses *network coding* yaitu dengan mengurangi informasi dari *relay* dengan *node tujuan*. Informasi baru di dapat kemudian di lakukan proses MRC sehingga *user 1* mendapatkan informasi *user 2*, begitupun sebaliknya. Alur desain sistemnya dapat dilihat pada gambar berikut ini:

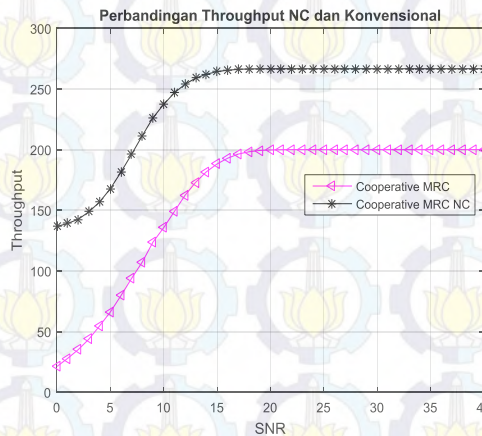


Gambar 3. 2 Desain Sistem Komunikasi Kooperatif *Network Coding*

Untuk kanal yang dibangkitkan pada sistem komunikasi kooperatif dengan dan tanpa *network coding* saat simulasi adalah kanal *Additive White Gaussian Noise* (AWGN). Sinyal yang melewati kanal tersebut akan ditambahkan nilai level amplitude noise pada level amplitude sinyal. Hasil simulasi Tugas Akhir ini dapat dilihat pada gambar berikut.



Gambar 3. 3 BER kooperatif tradisional dan *network coding*



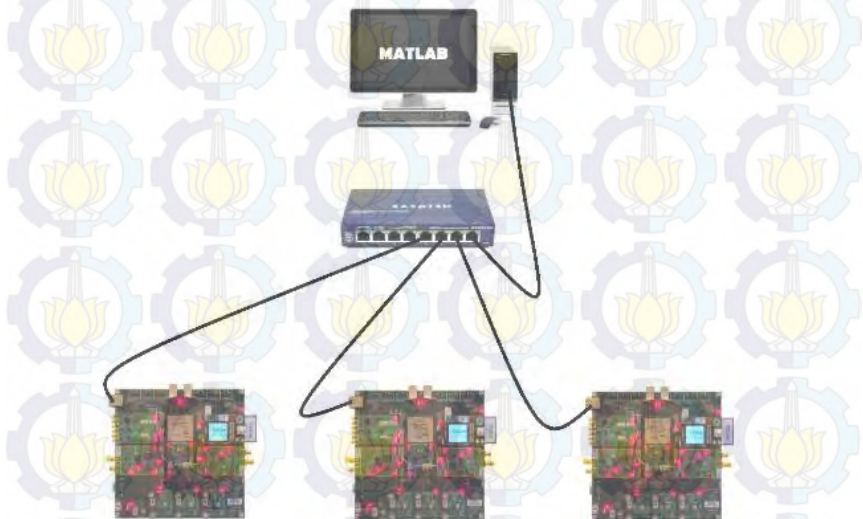
Gambar 3. 4 *Throughput* kooperatif tradisional dan *network coding*

Perbedaan antara sistem komunikasi kooperatif dengan skema *network coding* dan tradisional biasa dari sisi node *relay* adalah skema tradisional informasi yang diterima di node *relay* akan mengalami penguatan kemudian dikirimkan kembali ke tujuan tanpa melalui proses *decoding*. Sedangkan dengan skema *network coding* pada node *relay* dua

buah informasi yang diterima pada node *relay* dijumlahkan, kemudian di *broadcast* ke masing-masing *destination*

3.3 Integrasi PC dengan Modul WARP

Pada tugas akhir ini, Setelah melakukan desain sistem serta hasil, langkah selanjutnya yang dilakukan yaitu pengintegrasian antara PC dengan perangkat WARP. Perangkat WARP yang digunakan pada penelitian yaitu WARP V.2 dengan WARPLab versi 7. Konfigurasi pengintegrasian PC dengan perangkat WARP menggunakan *switch ethernet* dan kabel LAN. Integrasi antar PC dengan WARP ditunjukkan pada gambar 3.5.



Gambar 3. 5 Integrasi antara PC dengan WARP

Langkah – langkah pengintegrasian antara PC dengan WARP terdapat pada lampiran.

3.4 Pemodelan Sistem Komunikasi

Dalam melakukan pemodelan sistem komunikasi, maka harus ditentukan dulu beberapa parameter sistem komunikasi, serta bentuk sistem komunikasi transmisi sinyal sistem.

3.4.1 Parameter Sistem Komunikasi

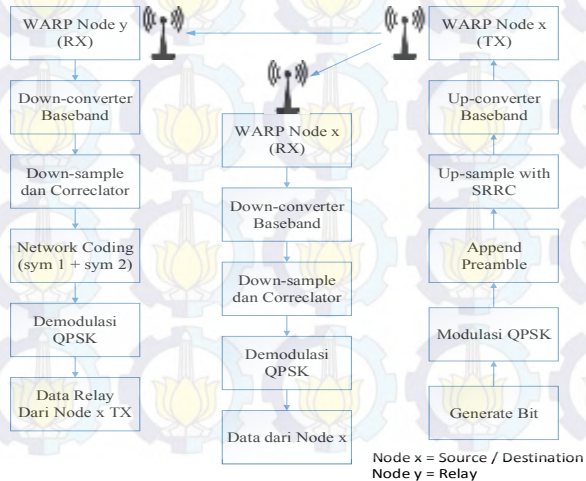
Pemodelan sistem komunikasi ini menggunakan beberapa parameter berdasarkan tabel 3.1.

Tabel 3. 1 Parameter Sistem Komunikasi

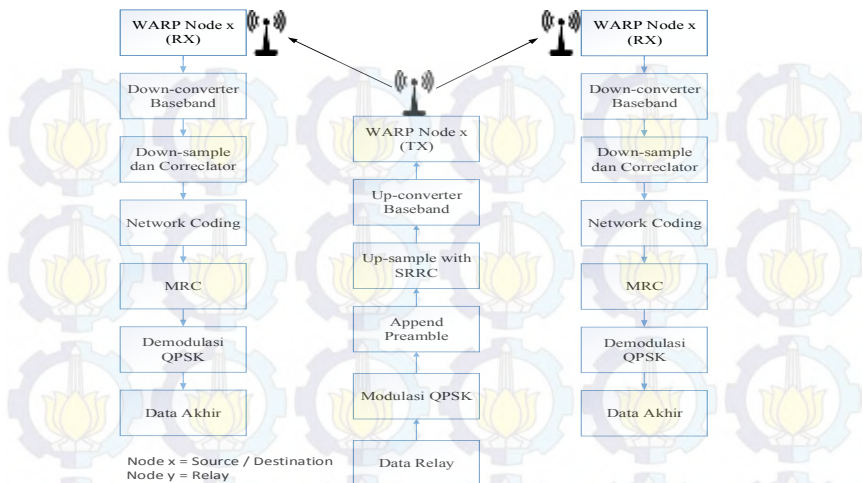
Parameter	Nilai
Modulasi	QPSK
Filter	SRRC
Filter order	6
<i>Rolloff factor</i>	0.9
<i>Preamble</i>	19
<i>Sample</i>	8
<i>Up converter dan down converter</i>	5 MHz
Frekuensi sampling	40 MHz
<i>Carrier channel</i>	[0:14]
<i>Transmitter baseband gain</i>	[0:3]
<i>Transmitter RF gain</i>	[0:63]
<i>Receiver basedband gain</i>	[0:31]

3.4.2 Sistem Komunikasi

Langkah-langkah komunikasi SIMO untuk skema *network coding* menggunakan WARP untuk mengirimkan data dapat ditunjukkan pada gambar 3.6.



Gambar 3. 6 Blok sistem komunikasi SIMO pada TS 1 atau 2



Gambar 3. 7 Blok diagram Sistem Komunikasi SIMO pada TS 3

Secara umum diagram blok sistem komunikasi kooperatif *network coding* ini terdiri atas *source*, *relay*, dan *destination*. Transmisi pada sistem terbagi menjadi tiga jenis transmisi, yaitu transmisi *source-relay* (S-R), transmisi *source-destination* (S-D) dan transmisi *relay-destination* (R-D). Pada data akhir dilakukan perhitungan BER dan *throughput*.

3.5 Implementasi Secara Realtime pada Modul WARP

Impelemntasi secara *relatime* pada modul WARP ini dilakukan setelah melakukan perancangan seperti konfigurasi perangkat dan pembedan sistem komunikasi. Untuk Penjelasan masing-masing blok diagram dari pemodelan sistem komunikasi adalah sebagai berikut:

3.5.1 Pada Pemancar

Informasi setiap blok dari gambar 3.6 dijelaskan sebagai berikut:

- a. *Generate bit*

Bit informasi akan dibangkitkan secara acak dimana kemungkinan kemunculan bit 1 dan 0 adalah *equiprobable*.

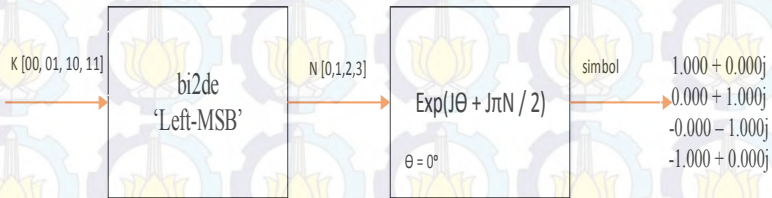
- b. Modulasi QPSK

Selanjutnya Bit yang dibangkitkan tersebut di *mapping* ke dalam simbol-simbol menggunakan teknik modulasi QPSK dengan $M = 4$ sehingga jumlah bit per simbol (k) yaitu 2. Hasil dari modulasi ini berupa simbol kompleks IQ (*Inphase Quadrature*). Pada modulasi QPSK setiap

dua bit informasi di *mapping* menjadi satu symbol, *mapping* bit informasi menjadi symbol-symbol dapat dilihat pada Gambar 3.8. Sementara untuk ketentuan urutan bit yaitu ‘*Left-MSB*’ seperti yang diperlihatkan pada Tabel 3.2.

Tabel 3. 2 *Mapping* Modulasi QPSK

Bit Info (left-msb)	Simbol	Nilai Simbol
00	S_1	$1.000 + 0.000j$
01	S_2	$0.000 + 1.000j$
10	S_3	$-0.000 - 1.000j$
11	S_4	$-1.000 + 0.000j$



Gambar 3. 8 Blok *Modulator* QPSK

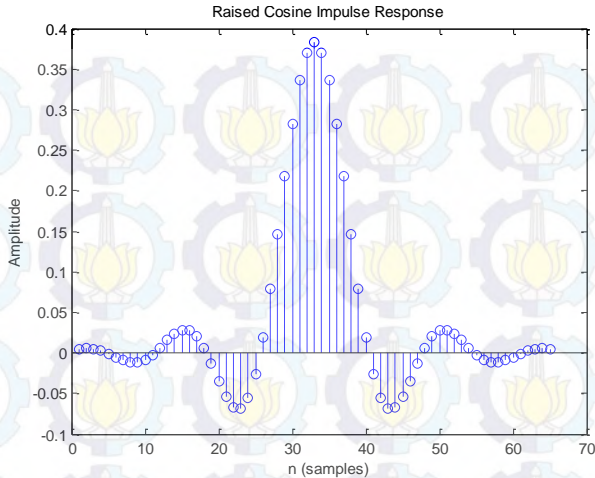
Berdasarkan Gambar 3.8, N merupakan bilangan desimal hasil konversi biner ke desimal dan θ adalah *phase offset*.

c. Preamble

Deretan simbol hasil modulasi tersebut akan ditambah dengan simbol pilot yang digunakan untuk respon estimasi kanal di penerima. Deretan simbol – simbol yang telah disisipi dengan simbol pilot, kemudian ditambah dengan *preamble*. *Preamble* merupakan deretan simbol yang digunakan sebagai sinkronisasi untuk mengetahui simbol pertama yang dikirimkan oleh *transmitter* pada bagian penerima. Penambahan simbol *preamble* ditunjukkan sebagai berikut.

d. Upsample Filter & Match-filter

Proses selanjutnya yaitu pembentukan pulsa dan upsampling. Proses pembentukan pulsa dan upsampling ini dilakukan dengan melewati filter *Square Root Raised Cosine* (SRRC) dengan *upsample rate* sebanyak 8 berdasarkan konfigurasi yang digunakan. Respon impulse dari filter *Square Root Raised Cosine* (SRRC) ditunjukkan pada gambar 3.9.



Gambar 3. 9 Respon impulse filter *squared root raised cosine*

Upsample & Filter merupakan suatu blok yang sering dikenal dengan pulse shapping pada sisi pemancar. Blok ini bertujuan menaikkan sample setiap simbol sesuai dengan jumlah sampling / symbol sehingga diperoleh sample sinyal baru dan kemudian di filter menggunakan filter pulse-shapping. Tujuan dari pulse shapping adalah merubah bentuk suatu sinyal ke bentuk sinyal lainnya agar diperoleh suatu bandwidth sinyal sesuai dengan yang diharapkan. Match-filter merupakan suatu filter yang diteapkan di penerima dengan jenis filter yang sama dengan pemancar. Match-filter bertujuan untuk mengurangi efek noise dari sinyal yang diterima.

e. Upconvert

Sebelum sinyal dikirimkan, sinyal akan di *upconvert* ke frekuensi 5 MHz. Proses *upconvert* ini dilakukan untuk menghindari pelemahan sinyal dari radio DC. Proses *upconvert* dilakukan dengan mengalikan sinyal terhadap frekuensi sinusoidal kompleks.

3.5.2 Proses Pengiriman Sinyal dan Komunikasi

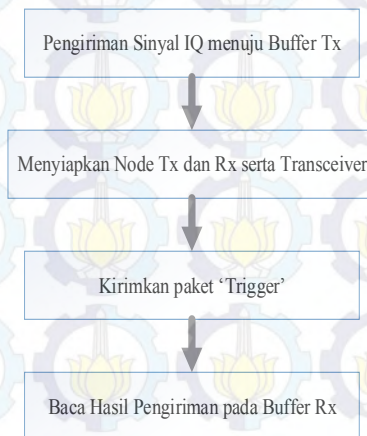
Proses pemrograman perangkat WARP dilakukan melalui WARPLab dan *software* MATLAB. Langkah-langkah komunikasi dengan menggunakan WARP untuk mengirimkan data seperti yang diperlihatkan pada Gambar 3.6 dan 3.7.

Gambar 3.10 menunjukkan proses pengiriman sinyal dan komunikasi pada WARP. Pada tahap awal, sinyal yang akan dikirim dalam bentuk kompleks IQ (*Inphase Quadrature*) dimana sinyal *inphase* merupakan bagian *real* sementara sinyal *quadrature* merupakan bagian *imaginer*. Sinyal yang akan dikirim disimpan pada *buffer* Tx dengan ukuran 2^{14} .

Setelah proses penyimpanan sinyal pada *buffer* Tx selesai, maka dilakukan persiapan pengiriman dan penerimaan data dengan mengaktifkan radio pada node pengirim dan penerima.

Proses komunikasi dilakukan dengan mengirimkan sebuah paket '*trigger*' kepada semua *node* secara bersamaan. Ketika sebuah *node* milik *transmitter* dalam kondisi *enable*, maka *node* tersebut akan memancarkan sinyal yang tadi terdapat pada *buffer*. Jika sebuah *node* milik *receiver* dalam kondisi *enable*, maka *node* tersebut akan menerima sinyal yang dipancarkan oleh *node transmitter*.

Ketika proses komunikasi selesai, data yang dikirimkan oleh *transmitter* akan diterima oleh *node* Rx. Proses pengambilan data dilakukan dengan membaca data yang terdapat pada *buffer* node Rx.



Gambar 3. 10 Proses Pengiriman Sinyal dan Komunikasi

3.5.3 Proses pada Penerima

Proses pada penerima adalah kebalikan dari proses pada pengirim, berikut adalah langkahnya :

a. *Downconvert*

Setelah bit diterima akan dilakukan proses *downconvert*, dimana pada proses ini sinyal akan dikembalikan ke frekuensi *baseband* semula. Pada proses ini sinyal akan dikalikan dengan frekuensi *upconvert* tanpa melakukan sinkronisasi fasa.

b. *Match Filter*

Setelah sinyal di *downconvert*, proses selanjutnya yaitu menghilangkan pengaruh sinyal yang tidak diperlukan seperti noise. Pada proses ini, frekuensi tinggi yang dihasilkan dari proses *downconvert* akan dihilangkan agar sesuai dengan bentuk pulsa *raised cosine* disisi pengirim.

c. *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*.

d. Proses *Network Coding*

Pada proses ini informasi berupa simbol IQ dari node relay yang dikurangkan dengan simbol IQ dari node source atau node destination.

e. *Combining*

Pada sistem komunikasi kooperatif aka ada 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 *maximum ratio combining* yaitu dengan menjumlahkan dua buah sinyal dari jalur relay dan siso, dan akan didapatkan data akhir.

f. Proses Demodulasi

Demodulasi adalah proses pengembalian dari symbol menjadi bit kembali.

3.6 Metode Pengukuran dan perhitungan BER dan *Throughput*

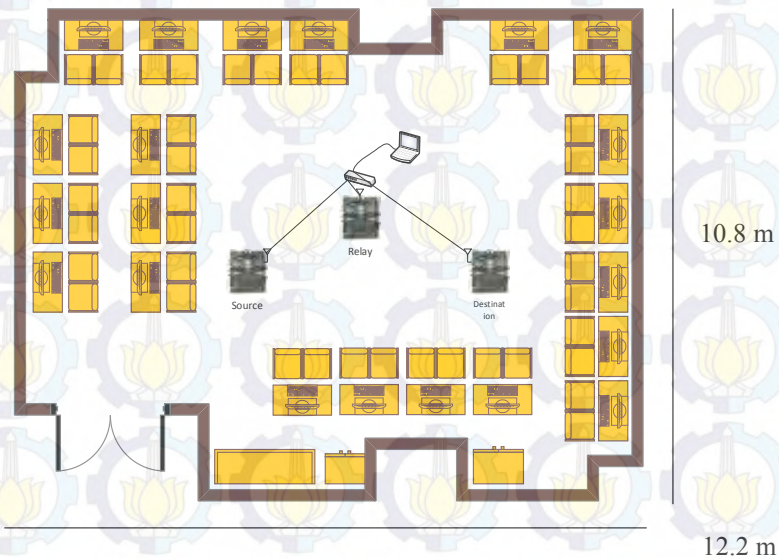
Pada tugas akhir ini, pengukuran dilakukan dalam kondisi lingkungan yang berbeda yaitu dalam lingkungan *indoor* dan lingkungan *outdoor*. Baik dalam kondisi lingkungan *indoor* maupun *outdoor* akan dilakukan pengukuran secara *line-of-sight* (LOS) / tanpa penghalang dan secara *non line-of-sight* (NLOS) / penghalang. Tujuan pengukuran pada lingkungan *indoor* dan *outdoor* adalah untuk membandingkan kinerja dari sistem komunikasi kooperatif *network coding* dengan kooperatif

tradisional menggunakan modul WARP dengan kondisi lingkungan yang berbeda.

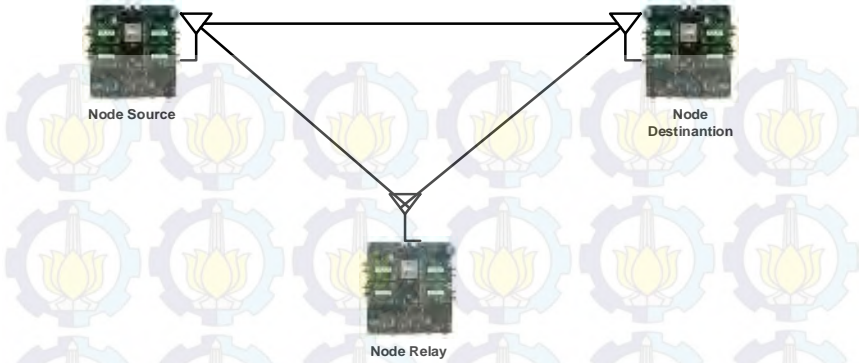
Pada pengukuran, terdapat kriteria – kriteria yang digunakan untuk mengetahui kinerja dari sistem komunikasi kooperatif *network coding* dengan kooperatif tradisional menggunakan modul WARP. Kriteria tersebut dapat dilihat pada Tabel 3.1. Dengan kriteria-kriteria yang ditentukan tersebut, hasil yang diharapkan dari pengukuran yaitu memastikan bahwa desain sistem mampu mengirimkan data pada kondisi ideal.

3.6.1 Pengukuran Lingkungan *Indoor*

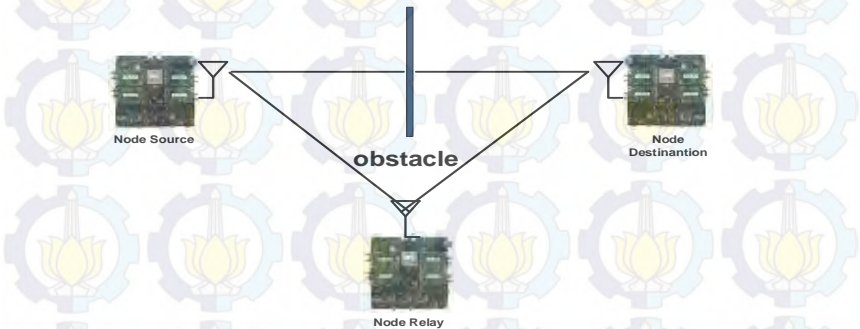
Pengukuran lingkungan *indoor* dilakukan di ruang AJ403 Laboratorium Pratikum Dasar Sistem Telekomunikasi jurusan Teknik Elektro. Jarak pengukuran antara WARP *node transmitter*, *node relay* dan WARP *node receiver* adalah 4m, 6m, dan 8m. Dengan jarak yang sama, pada kondisi tanpa penghalang (LOS) dan penghalang (NLOS). Kondisi LOS dan NLOS dapat ditunjukkan pada gambar 3.12 dan 3.13. Pada kondisi NLOS *Obstacle* yang digunakan berupa jaring kawat besi. Denah ruang pengukuran *indoor* ditunjukkan pada gambar 3.11.



Gambar 3. 11 Denah ruang AJ403 untuk pengukuran *indoor*



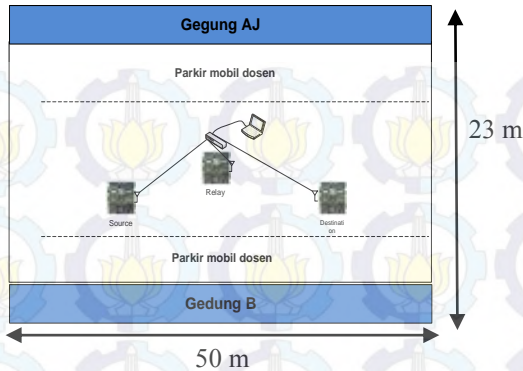
Gambar 3. 12 kondisi tanpa pengahalang



Gambar 3. 13 Kondisi dengan pengahalang

3.5.1 Pengukuran Lingkungan *Outdoor*

Pengukuran lingkungan *outdoor* dilakukan di area parkir mobil dosen Jurusan Teknik Elektro. Jarak pengukuran untuk *node source* dan *node destination* pada WARP disesuaikan dengan pengukuran pada kondisi *indoor* yaitu pada jarak 4m, 6m, dan 8m dengan *node Relay* tidak dirubah jaraknya untuk membandingkan kinerja dilingkungan *indoor* dan *outdoor*. Selain pada jarak 4m, 6m dan 8m, pengukuran pada lingkungan *outdoor* juga dilakukan pada jarak 12m, 18m dan 20m. Pengukuran lingkungan *outdoor* ini dilakukan dalam kondisi tanpa penghalang seperti gambar 3.12 dan kondisi dengan penghalang seperti gambar 3.13. Untuk denah pengukuran lingkungan *outdoor* ditunjukkan pada gambar 3.14.



Gambar 3. 14 Denah pengukuran di lingkungan *outdoor*

Parameter pengukuran BER lingkungan *outdoor* sama dengan parameter pengukuran BER lingkungan *indoor* seperti yang ditunjukkan pada tabel 3.1.

3.7 Perhitungan 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 [9].

$$\text{Bit Error Rate} = \frac{\text{Bit Diterima}}{\text{Bit Dikirim}}$$

3.8 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 [14] :

$$\frac{\text{Jumlah bit yang diterima}}{\text{waktu}} \text{ bps}$$



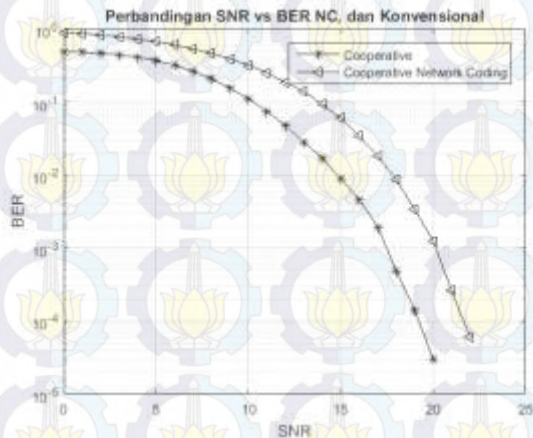
Halaman ini sengaja dikosongkan

BAB IV HASIL PENGUKURAN DAN ANALISIS DATA

4.1 Analisis Hasil Simulasi Sistem Komunikasi Kooperatif

Simulasi sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* dan tanpa *network coding* dilakukan menggunakan pemrograman Matlab. Kanal yang disimulasikan adalah kanal AWGN (*Additive White Gaussian Noise*). Untuk memastikan bahwa simulasi sistem yang dilakukan benar maka dilakukan proses validasi. Proses validasi dapat dilihat dari nilai BER. Parameter yang digunakan adalah jumlah bit yang dikirimkan sejumlah 16384 bit dengan modulasi *Quadrature Phase Shift Keying* (QPSK).

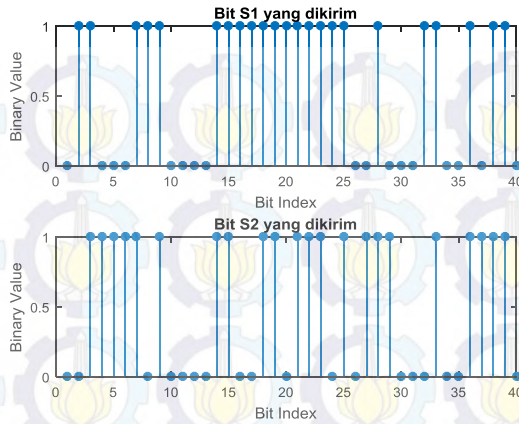
Berdasarkan hasil simulasi pada Gambar 4.1 BER sistem komunikasi kooperatif *amplify and forward* dengan skema tanpa NC menggunakan *combining* MRC lebih baik dari sistem komunikasi kooperatif dengan skema skema NC.



Gambar 4. 1 Simulasi sistem komunikasi kooperatif

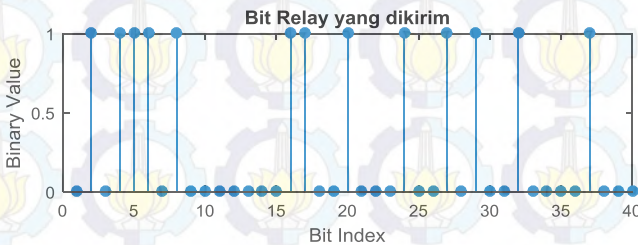
4.2 Output implementasi Sistem Komunikasi di Modul WARP

Random bit yang dibangkitkan pada node *source* 1 dan node *source* 2 ditunjukkan pada Gambar 4.2.



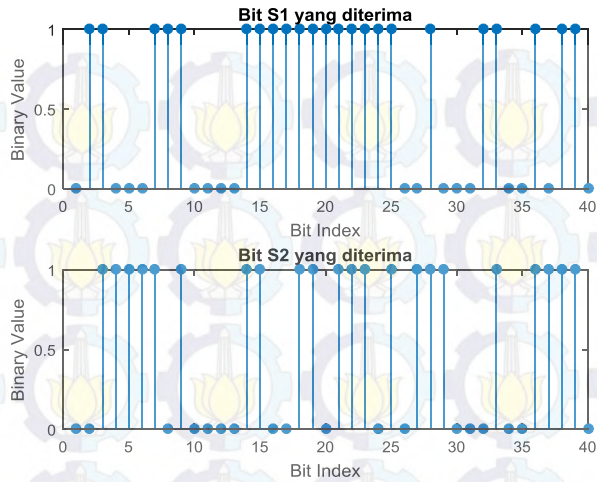
Gambar 4. 2 Random Bit Node Source 1 dan dan Node Source 2

Random bit tersebut selanjutnya dikirim ke node *relay*. Pada node *relay* ini informasi dari node *source* 1 dan node *source* 2 memasuki tahap *network coding*, dengan menjumlahkan simbol dari node *source* 1 dan 2 yang masih tercampur dengan *noise* (S_1n+S_2n). Simbol hasil penjumlahan tersebut dirubah dalam bentuk bit.



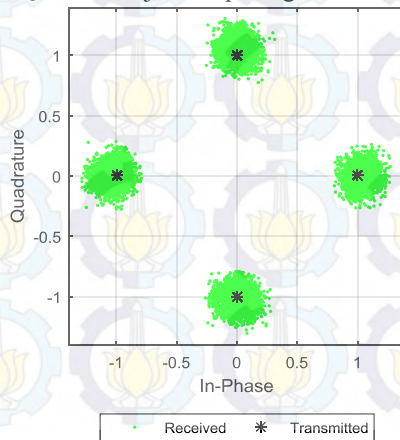
Gambar 4. 3 Bit yang diterima di Relay setelah tahap *Network Coding*

Informasi dari node *relay* tersebut di *broadcast* ke node *destination* 1 dan 2, pada node *destination* informasi dari node *source* 1 dan 2 akan mengalami proses *network coding* lagi dengan mengurangi dengan simbol informasi aslinya untuk mendapatkan informasi yang diinginkan, kemudian informasi tersebut di *combining* dan dirubah dalam bentuk bit.



Gambar 4. 4 Bit yang diterima pada node *Destination* 1 dan 2

Konstelasi dari simbol yang diterima dapat dilihat dengan menggunakan *scatterplot* sesuai dengan teknik modulasi QPSK yang digunakan. Simbol yang diamati adalah simbol yang telah melewati proses estimasi kanal dan perbaikan fasa. Konstelasi simbol dengan teknik modulasi QPSK ditunjukkan pada gambar 4.5.



Gambar 4. 5 Konstelasi simbol QPSK

4.3 Hasil Implementasi Sistem pada Modul WARP

Pada sub bab ini akan ditunjukkan hasil implementasi pada modul WARP dengan membandingkan antara sistem komunikasi kooperatif *amplify and forward* dengan skema NC dan tanpa NC, lalu membandingkan sistem komunikasi jalur langsung (SISO) dengan jalur dengan melewati *relay* (multihop), dan membandingkan kinerja sistem komunikasi NC berdasarkan jarak. Untuk menganalisa kinerja sistem komunikasi, parameter yang digunakan sebagai acuan adalah nilai *Bit Error Rate* (BER) dan *throughput*. Nilai BER ini dilihat sebagai dua fungsi yaitu sebagai fungsi daya pancar dan fungsi jarak. Dengan kata lain nilai BER ini dilihat dengan mengubah nilai daya yang dipancarkan oleh *source* dan mengubah jarak antara *source* ke *destination*. Adapun pengukurannya dilakukan di *indoor* (ruang AJ403) dan *outdoor* (lapangan parkir Jurusan Teknik Elektro ITS).

4.4 Analisis BER Hasil Pengukuran di Lingkungan Indoor

Pengukuran di lingkungan *indoor* ini dilakukan dengan 2 kondisi yaitu kondisi LOS (*Line of Sight*) seperti yang ditunjukkan pada gambar 3.12 dan kondisi NLOS (*Non Line of Sight*) seperti yang ditunjukkan pada dan 3.13, pengukuran dilakukan di ruang AJ403, Laboratorium Pratikum Dasar Sistem Telekomunikasi Jurusan Teknik Elektro. Baik dalam kondisi LOS dan NLOS ini dilakukan dengan mengubah jarak antara *node* pengirim dan *node* penerima yaitu pada jarak 4m, 6m dan 8m untuk mendapatkan nilai *Bit Error Rate* (BER). Analisis hasil pengukuran akan dibandingkan antara sistem kooperatif skema tradisional dan dengan skema *network coding* dan membandingkan kinerja jalur SISO dan multihop. Jumlah bit yang dikirimkan yaitu 16384 dan parameter gain yang digunakan untuk pengukuran yaitu:

- Tx Baseband gain (Tx_BB) : 2
- Tx Baseband gain (Tx_BB) relay : 3
- Tx Radio Frequency gain (Tx_Rf) : 0:4:64
- Rx Baseband gain (Rx_BB) : 1
- Rx Radio Frekuensi gian (Rx_Rf) : 1

4.4.1 Analisa Pengukuran Indoor Kondisi LOS

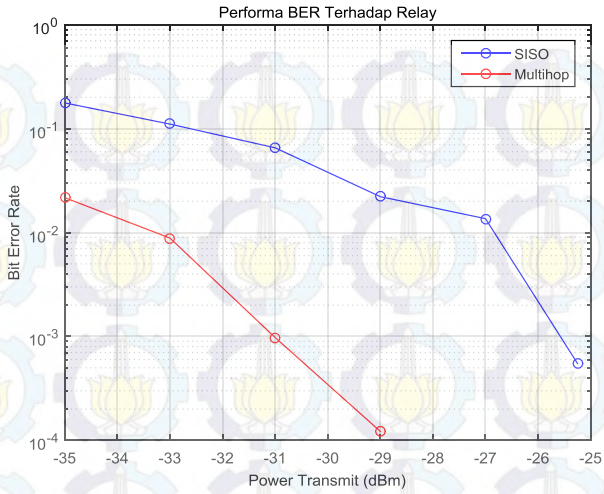
Analisis Pengukuran di lingkungan *indoor* dengan kondisi LOS sesuai gambar 3.12 dilakukan dengan mengubah fungsi level daya pancar (Tx_Rf) dari 0 sampai 60 pada jarak 4m, 6m, dan 8m. Hasil dari pengukuran dengan mengubah level daya pancar (Tx_Rf) di lingkungan

indoor dengan kondisi LOS ditunjukkan pada tabel 4.1, tabel 4.2 dan tabel 4.3.

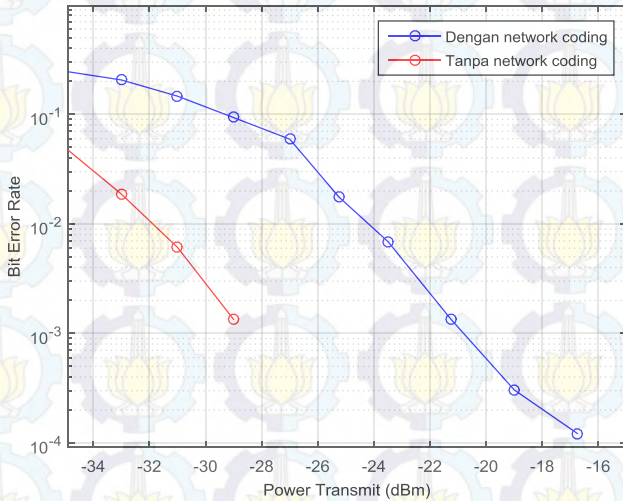
Tabel 4. 1 BER pada jarak 4 meter (indoor) kondisi LOS

Gain	Ptx (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
0	-35	0.1769	0.0217	0.2474	0.0496
4	-33	0.1107	0.0089	0.2063	0.0187
8	-31	0.0654	9.7656e-4	0.1467	0.0061
12	-29	0.0222	1.2207e-4	0.0935	0.0013
16	-27	0.0136	0	0.0594	0
20	-25.25	5.4932e-4	0	0.0176	0
24	-23.5	0	0	0.0069	0
28	-21.25	0	0	0.0013	0
32	-19	0	0	3.0518e-4	0
36	-16.75	0	0	1.2207e-4	0
40	-14.5	0	0	0	0
44	-13	0	0	0	0
48	-11.5	0	0	0	0
52	-9.25	0	0	0	0
56	-7	0	0	0	0
60	-4.7143	0	0	0	0

Dari nilai tabel 4.1 dapat dibuat kurva BER jalur multihop dengan jalur SISO dan kurva BER sistem komunikasi kooperatif *amplify and forward* dengan skema NC dan tanpa NC dengan fungsi level gain daya pancar (Tx_Rf) pada lingkungan *indoor* dengan kondisi LOS seperti pada gambar 4.6 dan gambar 4.7.



Gambar 4. 6 BER untuk jalur komunikasi multihop dan SISO



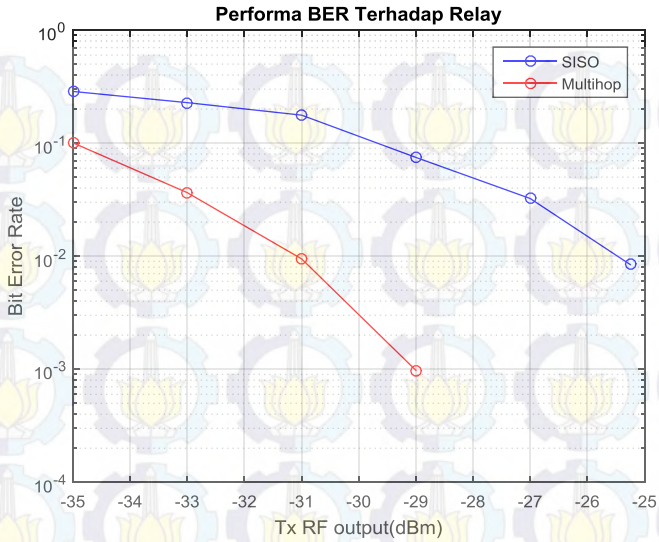
Gambar 4. 7 BER perancangan sistem dengan NC dan tanpa NC

Dari gambar 4.6 nilai BER untuk jalur SISO dan multihop semakin kecil seiring dengan kenaikan level gain daya pancar. Hal ini disebabkan semakin besar daya yang dipancarkan dengan noise yang sama, maka lebih tahan terhadap noise dan informasi yang dikirim semakin sedikit terjadi kesalahannya. Analisis sistem komunikasi kooperatif *amplify and forward* dengan skema NC untuk jalur SISO bernilai BER $5.4932e-4$ pada level gain 20, ketika level gain dinaikkan menjadi 21 nilai dari BER SISO menjadi nol. Sedangkan untuk jalur multihop ketika level gain 16 nilai BER sudah nol.

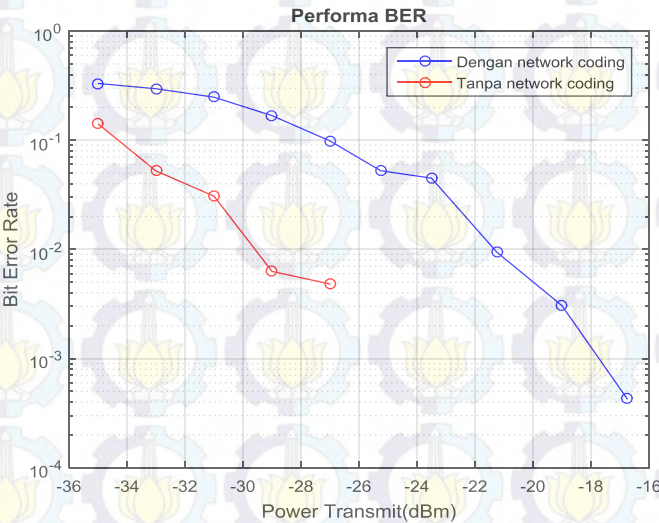
Gambar 4.7 menjelaskan sistem komunikasi kooperatif dengan skema *network coding* gain 40 nilai BER sudah nol, untuk skema tanpa NC ketika berada pada level gain 20 nilai BER sudah nol.

Tabel 4. 2 BER pada jarak 6 meter (indoor) kondisi LOS

Gain	Ptx (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
0	-35	0.28638	0.10046	0.3316	0.14166
4	-33	0.22797	0.036438	0.29657	0.052368
8	-31	0.17676	0.0094604	0.24829	0.030762
12	-29	0.074707	0.0009765	0.16821	0.006286
16	-27	0.032349	0	0.098572	0.004821
20	-25.25	0.008423	0	0.052551	0
24	-23.5	0	0	0.044983	0
28	-21.25	0	0	0.009460	0
32	-19	0	0	0.003052	0
36	-16.75	0	0	0.000427	0
40	-14.5	0	0	0	0
44	-13	0	0	0	0
48	-11.5	0	0	0	0
52	-9.25	0	0	0	0
56	-7	0	0	0	0
60	-4.7143	0	0	0	0



Gambar 4. 8 BER untuk jalur komunikasi multihop dan SISO



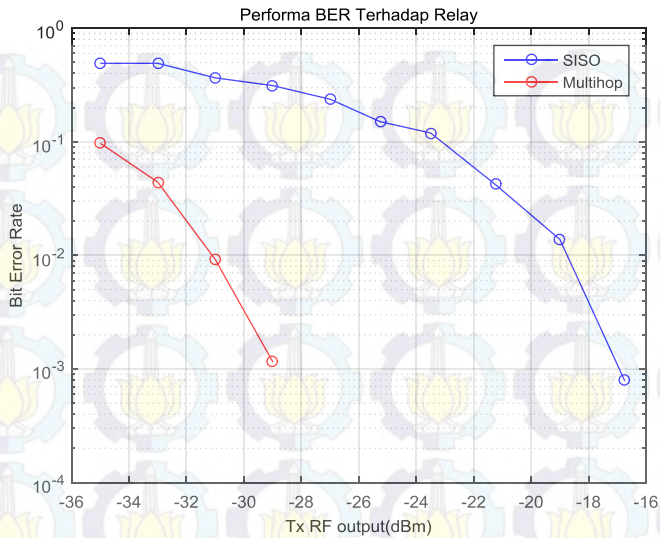
Gambar 4. 9 BER perancangan sistem dengan NC dan tanpa NC

Dari gambar 4.8, untuk jalur komunikasi SISO pada jarak 6 meter ini, BER bernilai nol ketika diberikan gain sebesar 24 atau Ptx -23.5 dBm, sedangkan untuk jalur multihop BER bernilai nol ketika gain hanya bernilai 16 atau power transmit ketika -27 dBm

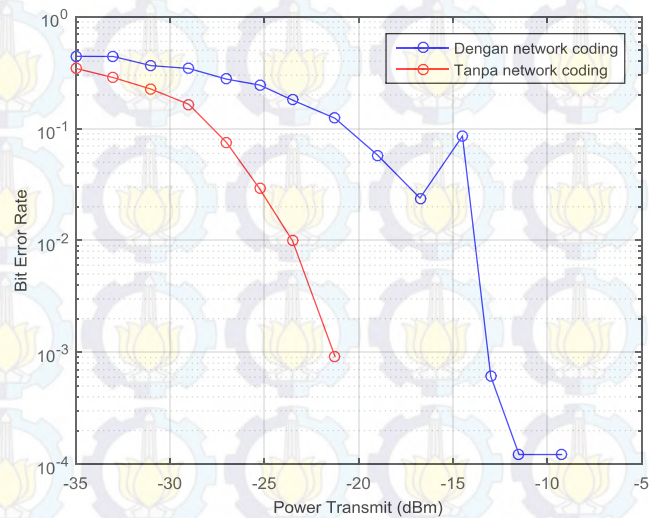
Untuk skema sistem komunikasi kooperatif *amplify and forward* dengan NC dan tanpa NC, pada skema NC pada level gain 40 atau Ptx -14.5 dBm sudah mendapatkan nilai BER nol, dan untuk skema tanpa NC BER bernilai nol ketika level gain 20 atau Ptx -25.25 dBm.

Tabel 4. 3 BER pada jarak 8 meter (indoor) kondisi LOS

Gain	Ptx (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
0	-35	0.4903	0.097168	0.44232	0.34546
4	-33	0.49109	0.044128	0.4408	0.28546
8	-31	0.36578	0.0092163	0.36511	0.22528
12	-29	0.31274	0.0011597	0.34454	0.16461
16	-27	0.237	0	0.27655	0.075012
20	-25.25	0.15021	0	0.24506	0.028992
24	-23.5	0.11963	0	0.1795	0.009887
28	-21.25	0.042114	0	0.12347	0.000915
32	-19	0.013794	0	0.057922	0
36	-16.75	0.0007934	0	0.023926	0
40	-14.5	0	0	0.085632	0
44	-13	0	0	0.0006103	0
48	-11.5	0	0	0.0001220	0
52	-9.25	0	0	0.0001220	0
56	-7	0	0	0	0
60	-4.7143	0	0	0	0



Gambar 4. 10 BER untuk jalur komunikasi multihop dan SISO



Gambar 4. 11 BER perancangan sistem dengan NC dan tanpa NC

Dari gambar 4.10, untuk jalur komunikasi SISO pada jarak 8 meter ini, BER bernilai nol ketika diberikan gain sebesar 40 atau Ptx -14.5 dBm, sedangkan untuk jalur multihop BER bernilai nol ketika gain hanya 16 atau power transmit ketika -27 dBm Untuk skema sistem komunikasi kooperatif *amplify and forward* dengan NC dan tana NC berdasarkan gambar 4.11, pada skema NC pada level gain 56 atau Ptx -7 dBm sudah mendapatkan nilai BER nol, dan untuk skema tradisional BER bernilai nol ketika level gain 32 atau Ptx -19 dBm.

Berdasarkan gambar diatas, kondisi lingkungan *indoor* dengan jarak 4 meter, 6 meter dan 8 meter dapat ditarik kesimpulan bahwa untuk lingkungan indoor dengan kondisi LOS, untuk jalur komunikasi multihop lebih bagus dibandingkan dengan komunikasi SISO dilihat dari segi parameter BER. Sedangkan untuk performasi sistem komunikasi kooperaif dengan skema *network coding* dan tanpa NC berdasarkan BER, skema tanpa NC lebih bagus dibandingkan dengan skema *network coding*. Hal ini dikarenakan pada skema *network coding* nya menggunakan skema PNC, yaitu simbol dari user 1 yang terkena *noise* dan simbol user 2 yang terkena *noise* dijumlahkan sebelum proses *decoding* dan demodulasi.

4.4.2 Analisa Pengukuran Indoor Kondisi NLOS

Pengukuran di lingkungan *indoor* dengan kondisi NLOS ini sama dengan kondisi LOS, perbedaanya kondisi NLOS ini sesuai skema gambar 3.13. Parameter yang digunakan juga sama dengan kondisi LOS lingkungan *indoor*.

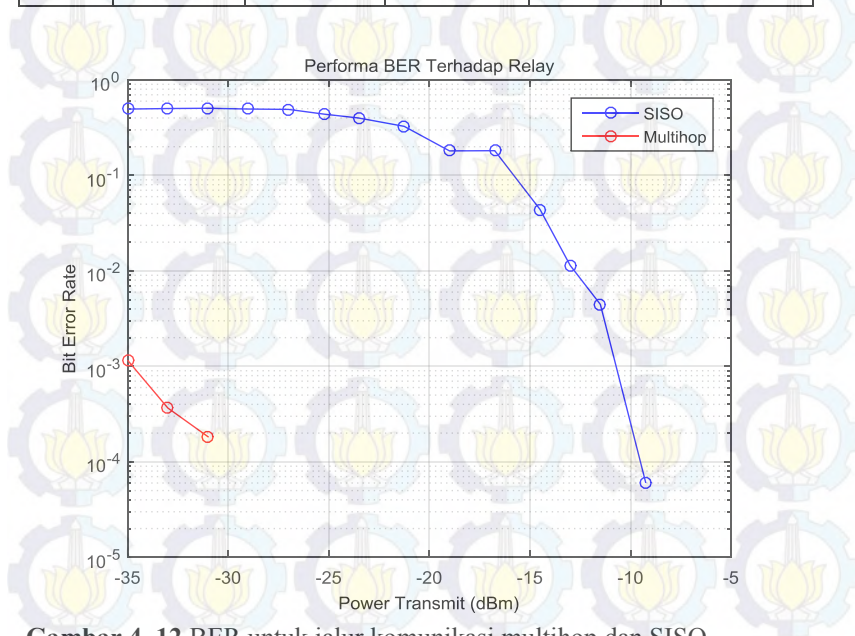
Hasil dari pengukuran dengan mengubah level daya pancar (Tx Rf) di lingkungan indoor dengan kondisi NLOS ditunjukkan pada tabel 4.4, tabel 4.5 dan tabel 4.6

Tabel 4. 4 BER pada jarak 4 meter (indoor) kondisi NLOS

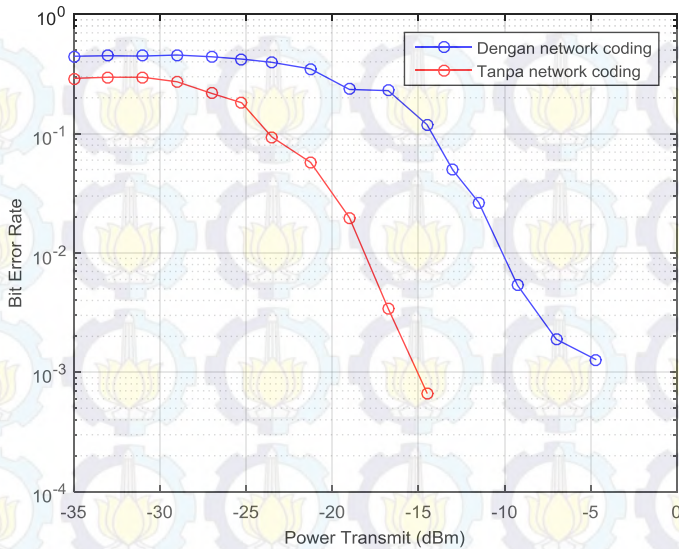
Gain	Ptx (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
0	-35	0.49548	0.0011597	0.4455	0.29071
4	-33	0.50214	0.0003662	0.44971	0.297
8	-31	0.50519	0.0001831	0.44696	0.29706
12	-29	0.49603	0	0.45587	0.27216
20	-25.25	0.43903	0	0.42157	0.18146

Tabel 4.4 BER pada jarak 4 meter (indoor) kondisi NLOS
(Tabel lanjutan)

Gain	P _{tx} (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
28	-21.25	0.32666	0	0.34882	0.057373
32	-19	0.18073	0	0.23468	0.019775
36	-16.75	0.18164	0	0.22937	0.003418
40	-14.5	0.044067	0	0.11908	0.000671
44	-13	0.011414	0	0.049438	0
48	-11.5	0.004456	0	0.02655	0
52	-9.25	6.104e-05	0	0.005371	0
56	-7	0	0	0.001892	0
60	-4.7143	0	0	0.001282	0



Gambar 4. 12 BER untuk jalur komunikasi multihop dan SISO



Gambar 4.13 BER perancangan sistem dengan NC dan tanpa NC

Dari gambar 4.12, untuk jalur komunikasi SISO pada jarak 4 meter dengan kondisi NLOS, BER bernilai nol pada level gain 56 atau pada Ptx -7 dBm, sedangkan untuk jalur multihop BER bernilai nol ketika level gain 12 atau power transmit ketika -29 dBm.

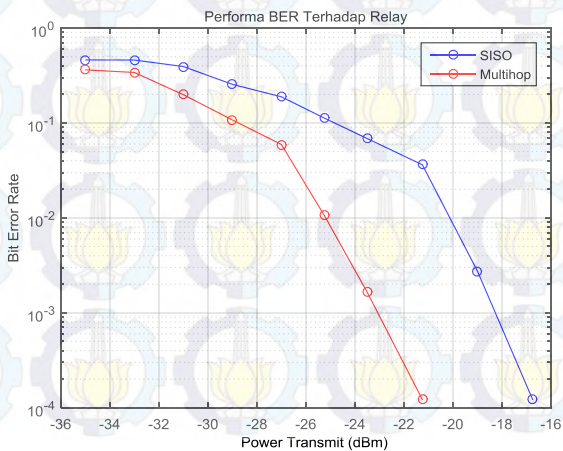
Untuk skema sistem komunikasi kooperatif *amplify and forward* dengan *network coding* dan tanpa *network coding* berdasarkan gambar 4.13, skema tana NC BER bernilai nol ketika level gain 44 atau dengan Ptx -13 dBm. Sedangkan untuk skema *network coding* pada level gain tertinggi yaitu 60 atau pada *power transmit* -4.7143 masih terjadi kesalahan dalam bit yang diterima. yaitu dengan BER 0.001282.

Tabel 4. 5 BER pada jarak 6 meter (indoor) kondisi NLOS

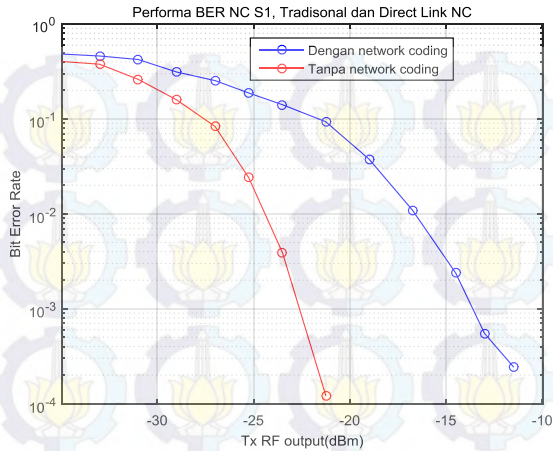
Gain	Ptx (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
0	-35	0.46234	0.36304	0.48444	0.40442
4	-33	0.46094	0.34094	0.4613	0.37793

Tabel 4.5 BER pada jarak 6 meter (indoor) kondisi NLOS
(Tabel lanjutan)

Gain	P _{tx} (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
12	-29	0.25549	0.10773	0.31281	0.15924
16	-27	0.1889	0.059387	0.25397	0.083374
20	-25.25	0.11273	0.01062	0.18799	0.024048
24	-23.5	0.068604	0.0016479	0.14148	0.003906
28	-21.25	0.036316	0.0001221	0.093567	0.000122
32	-19	0.002747	0	0.03772	0
36	-16.75	0.000122	0	0.010986	0
40	-14.5	0	0	0.002380	0
44	-13	0	0	0.000549	0
48	-11.5	0	0	0.000244	0
52	-9.25	0	0	0	0
56	-7	0	0	0	0
60	-4.7143	0	0	0	0



Gambar 4. 14 BER untuk jalur komunikasi multihop dan SISO



Gambar 4. 15 BER perancangan sistem dengan NC dan tanpa NC

Dari gambar 4.14, untuk jalur komunikasi SISO pada jarak 6 meter dengan kondisi NLOS, BER bernilai nol pada level gain 40 atau pada P_{tx} -14.5 dBm, sedangkan untuk jalur multihop BER bernilai nol ketika level gain 32 atau power transmit ketika -19 dBm.

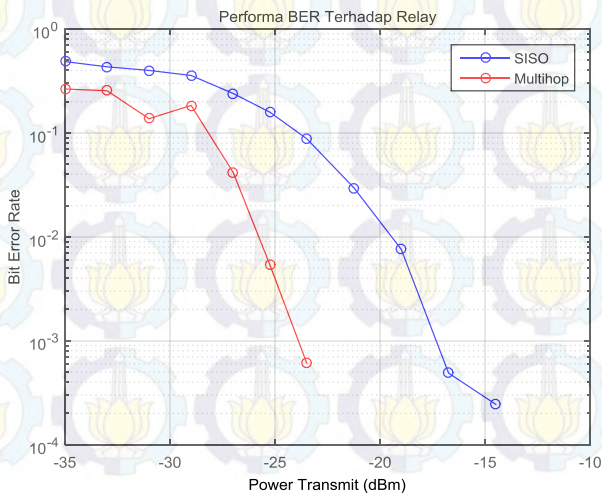
Untuk skema sistem komunikasi kooperatif *amplify and forward* dengan *network coding* dan tanpa *network coding* berdasarkan gambar 4.15, skema tanpa *network coding* BER bernilai nol ketika level gain 44 atau dengan P_{tx} -13 dBm, namun ketika level gain berada pada level tertinggi yaitu 60 atau pada *power transmit* -4.7143 terjadi kesalahan dalam bit yang diterima., nilai BER untuk skema tradisional ini 0.041138. Hal ini dikarenakan karena pengaruh kanal. Sedangkan untuk skema *network coding* pada level gain 48 atau pada *power transmit* -11.5 dBm BER sama dengan nol.

Tabel 4. 6 BER pada jarak 8 meter (indoor) kondisi NLOS

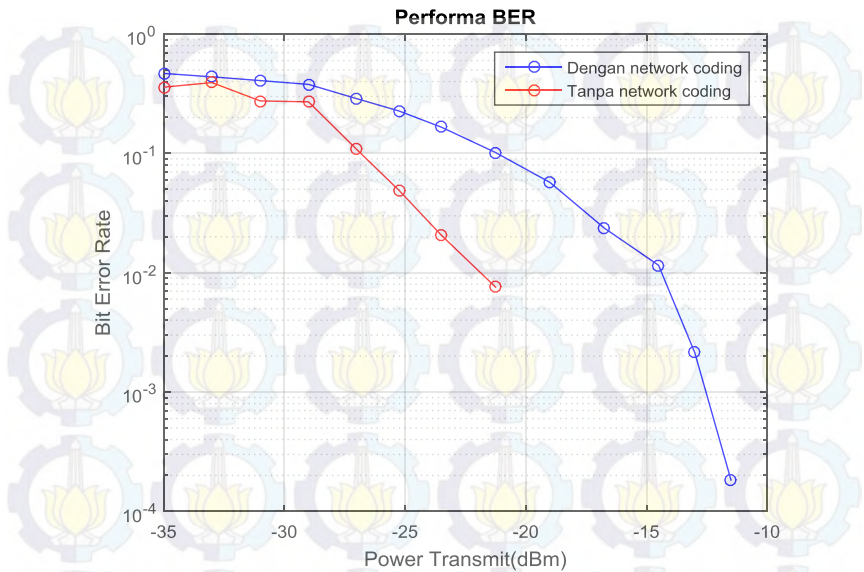
Gain	Ptx (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
0	-35	0.48621	0.26465	0.46771	0.35883
4	-33	0.43262	0.2547	0.43799	0.39197
12	-29	0.35724	0.18402	0.37854	0.27032

Tabel 4.6 BER pada jarak 8 meter (indoor) kondisi NLOS
(Tabel lanjutan)

Gain	P _{tx} (dBm)	BER (Bit Error Rate)			
		SISO	Multihop	NC	Tanpa NC
16	-27	0.23981	0.041748	0.28741	0.10779
20	-25.25	0.15826	0.0053711	0.22656	0.04895
24	-23.5	0.088013	0.0006104	0.16595	0.02063
28	-21.25	0.029602	0	0.10162	0.007568
32	-19	0.007568	0	0.057129	0
36	-16.75	0.000488	0	0.023682	0
40	-14.5	0.000244	0	0.011536	0
44	-13	0	0	0.002136	0
48	-11.5	0	0	0.000183	0
52	-9.25	0	0	0	0
56	-7	0	0	0	0
60	-4.7143	0	0	0	0



Gambar 4. 16 BER untuk jalur komunikasi multihop dan SISO



Gambar 4. 17 BER perancangan sistem dengan NC dan tanpa NC

Dari gambar 4.16, untuk jalur komunikasi SISO pada jarak 8 meter dengan kondisi NLOS, BER bernilai nol pada level gain 44 atau pada Ptx -13 dBm, sedangkan untuk jalur multihop BER bernilai nol ketika level gain 28 atau power transmit ketika -21.25 dBm.

Untuk skema sistem komunikasi kooperatif *amplify and forward* dengan *network coding* dan tanpa NC berdasarkan gambar 4.17, skema tanpa NC BER bernilai nol ketika level gain 32 atau dengan Ptx -19 dBm. Sedangkan untuk skema *network coding* pada level gain 52 atau dengan *power transmit* -9.25 dBm.

Berdasarkan gambar diatas, kondisi lingkungan *indoor* dengan jarak 4 meter, 6 meter dan 8 meter dapat ditarik kesimpulan bahwa untuk lingkungan indoor dengan kondisi NLOS, BER untuk jalur komunikasi multihop lebih bagus dibandingkan dengan komunikasi SISO, sama halnya pada konsisi LOS, Sedangkan untuk performasi BER sistem komunikasi kooperaif dengan skema *network coding* dan tanpa *network coding* berdasarkan gambar diatas skema tanpa *network coding* lebih bagus dibandingkan dengan skema *network coding*.

4.5 Analisis BER Hasil Pengukuran di Lingkungan *Outdoor*

Proses pengukuran di lingkungan *outdoor* dilakukan di area parkir mobil dosen Jurusan Teknik Elektro. Pengukuran di lingkungan *outdoor* ini dilakukan dengan 2 kondisi yaitu kondisi LOS (*Line of Sight*) dan kondisi NLOS (*Non Line of Sight*). Pengukuran dilakukan dengan mengubah fungsi jarak, baik kondisi LOS maupun NLOS dilakukan dengan mengubah jarak antara *node source* dan *node destination* yaitu pada jarak 4m, 6m, 8m, 12m, 18m dan 24m dan 30m untuk mendapatkan nilai *Bit Error Rate* (BER). Pengubahan jarak ini dilakukan untuk mengetahui sejauh mana sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* bekerja. Pengukuran di lingkungan *outdoor* ini juga dilakukan dengan mengubah *level gain* pemancar pada jarak yang sama untuk mendapatkan nilai BER. Jumlah bit yang dikirimkan yaitu 16384 bit dan parameter gain yang digunakan untuk pengukuran yaitu:

- Tx *Baseband gain* (Tx_BB) : 2
- Tx *Baseband gain* (Tx_BB) relay : 3
- Tx *Radio Frequency gain* (Tx_Rf) : 0:4:64
- Rx *Baseband gain* (Rx_BB) : 1
- Rx *Radio Frekuensi gian* (Rx_Rf) : 1

4.5.1. Analisis Pengukuran *Outdoor* kondisi LOS

Berdasarkan data pada tabel 4.7 hasil pengukuran menunjukkan seiring kenaikan nilai *level gain* daya pancar, nilai BER untuk sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* semakin kecil.

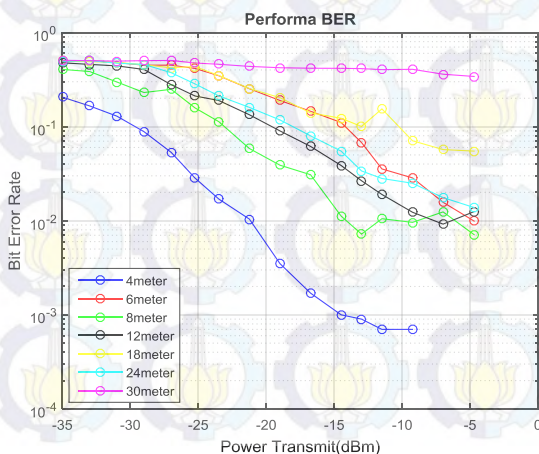
Tabel 4. 7 BER sistem komunikasi kooperatif dengan skema *network coding*

Ptx (dBm)	Jarak (BER)						
	4m	6m	8m	12m	18m	24m	30m
-35	0.2054	0.4916	0.4053	0.4767	0.5016	0.4936	0.5065
-33	0.1658	0.5027	0.3848	0.4579	0.4898	0.5029	0.5022
-31	0.1292	0.4786	0.2978	0.4420	0.4811	0.4718	0.4965
-29	0.0883	0.4493	0.2302	0.4047	0.4547	0.4576	0.5023
-25.25	0.0284	0.4151	0.1580	0.2115	0.4340	0.2842	0.4739

Tabel 4.7 BER sistem komunikasi kooperatif dengan skema *network coding* (Tabel lanjutan)

P _{tx} (dBm)	Jarak (BER)						
	4m	6m	8m	12m	18m	24m	30m
-23.5	0.0170	0.3472	0.1129	0.1894	0.3491	0.2121	0.4595
-21.25	0.0104	0.2527	0.0585	0.1336	0.2538	0.1600	0.4362
-19	0.0035	0.1894	0.0389	0.0900	0.2030	0.1177	0.4222
-16.75	0.0017	0.1456	0.0308	0.0616	0.1398	0.0798	0.4171
-14.5	0.0010	0.1105	0.0112	0.0386	0.1227	0.0549	0.4160
-13	0.0009	0.0677	0.0072	0.0266	0.1002	0.0338	0.4132
-11.5	0.0007	0.0355	0.0105	0.0190	0.1542	0.0281	0.4030
-9.25	0.0007	0.0285	0.0096	0.0123	0.0707	0.0250	0.4062
-7	0	0.0157	0.0124	0.0092	0.0570	0.0176	0.3603
-4.7143	0	0.0099	0.0071	0.0125	0.0549	0.0138	0.3409

Dari nilai tabel 4.7 dapat dibuat kurva BER sistem komunikasi kooperatif dengan skema *network coding* pada lingkungan *outdoor* dengan kondisi LOS seperti pada gambar 4.18.



Gambar 4. 18 BER sistem terhadap fungsi jarak dan P_{tx} skema NC

Berdasarkan tabel 4.7 dan gambar 4.18, nilai BER kooperatif *amplify and forward* dengan skema *network coding* pada kondisi LOS dengan jarak yang berbeda-beda dan power transmit yang berbeda-beda juga semakin kecil seiring dengan kenaikan level gain daya pancar. Analisis sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* kondisi LOS pada jarak 4 meter memiliki performansi yang bagus berdasarkan nilai BER nya dibandingkan dengan jarak 30 meter atau jarak lainnya. Pada jarak 4 meter dengan *power transmit* -7 dBm BER sudah bernilai nol, sedangkan untuk kondisi jarak lainnya pada level *power transmit* tertinggi masih ada terjadi kesalahan dalam bit yang diterima. Hal ini bisa terjadi karena pengaruh kondisi LOS (*Line of Sight*)

4.5.2. Analisis Pengukuran *Outdoor* kondisi NLOS

Hasil dari pengukuran dengan mengubah level daya pancar (Tx_Rf) dan jarak di lingkungan *outdoor* dengan kondisi NLOS ditunjukkan pada tabel 4.8.

Tabel 4. 8 BER sistem komunikasi kooperatif dengan skema *network coding*

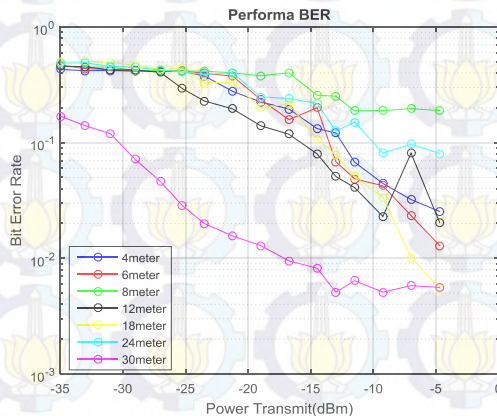
P _{tx} (dBm)	Jarak (BER)						
	4m	6m	8m	12m	18m	24m	30m
-35	0.4297	0.4664	0.4625	0.4597	0.4977	0.4859	0.1697
-33	0.4195	0.4371	0.4496	0.4536	0.5053	0.4912	0.1400
-31	0.4202	0.4285	0.4502	0.4247	0.4934	0.4563	0.1196
-29	0.4155	0.4183	0.4205	0.4258	0.4677	0.4489	0.0715
-27	0.4127	0.4172	0.4204	0.4048	0.4292	0.4245	0.0460
-25.25	0.4126	0.4187	0.4165	0.2932	0.4395	0.4092	0.0286
-23.5	0.3777	0.3972	0.4153	0.2292	0.3297	0.3892	0.0200
-21.25	0.2781	0.3762	0.4017	0.1962	0.3309	0.4025	0.0156
-19	0.2250	0.2368	0.3795	0.1400	0.2004	0.2487	0.0129
-16.75	0.1956	0.1580	0.4014	0.1183	0.2159	0.2413	0.0095
-14.5	0.1323	0.2007	0.2570	0.0800	0.1068	0.2182	0.0082
-11.5	0.0677	0.0487	0.1884	0.0408	0.0510	0.1504	0.0064

Tabel 4.8 BER sistem komunikasi kooperatif dengan skema *network coding* (Tabel lanjutan)

Ptx (dBm)	Jarak (BER)						
	4m	6m	8m	12m	18m	24m	30m
-9.25	0.0444	0.0424	0.1895	0.0230	0.0338	0.0813	0.0051
-7	0.0322	0.0234	0.1966	0.0812	0.0101	0.0970	0.0058
-4.7143	0.0254	0.0129	0.1895	0.0125	0.0055	0.0806	0.0056

Dari nilai tabel 4.8 dapat dibuat kurva BER sistem komunikasi kooperatif dengan skema *network coding* pada lingkungan *outdoor* dengan kondisi NLOS seperti pada gambar 4.19.

Berdasarkan tabel 4.8 dan gambar 4.19, nilai BER kooperatif *amplify and forward* dengan skema NC pada kondisi NLOS dengan jarak yang berbeda-beda dan power transmit yang berbeda-beda juga semakin kecil seiring dengan kenaikan level gain daya pancar. Analisis sistem komunikasi kooperatif *amplify and forward* dengan skema NC kondisi NLOS pada jarak 30 meter memiliki performasi yang bagus berdasarkan nilai BER nya dibandingkan dengan jarak lainnya. Pada semua jarak dengan level Ptx tertinggi masih terjadi kesalahan dalam proses penerimaan bit dipenerima. Dibandingkan dengan semua jarak, pada jarak 30 meter kesalahan bit (BER) pada level *power transmit* tertinggi bernilai 0.0056. Hal ini bisa terjadi karena pengaruh kondisi NLOS (*Line of Sight*)



Gambar 4. 19 BER sistem terhadap fungsi jarak dan Ptx skema NC

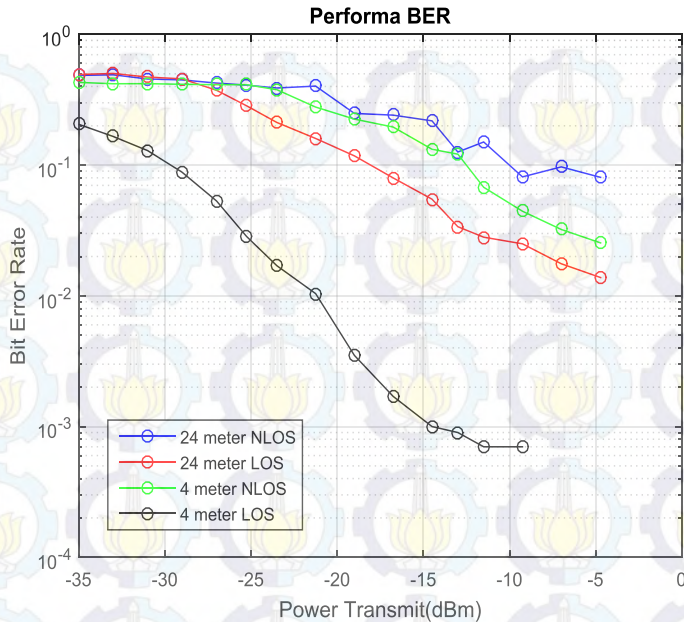
4.5.3. Analisis Pengukuran *Outdoor* kondisi LOS dan NLOS

Analisis hasil pengukuran dilingkungan *outdoor* dengan kondisi LOS dan NLOS membandingkan data pada kondisi jarak yang sama untuk sistem komunikasi kooperatif dengan skema *network coding*. Pada pengukuran ini dilakukan analisis pada jarak 24 meter. Hasil dari pengukuran dengan mengubah level daya pancar pada jarak 24 meter ditunjukkan pada tabel 4.6 dan gambar 4.6.

Tabel 4. 9 BER kondisi LOS dan NLOS pada jarak 4 dan 24 meter

Ptx (dBm)	Jarak 4 meter		Jarak 24 meter	
	LOS	LOS	LOS	NLOS
-35	0.4936	0.4936	0.2054	0.4297
-33	0.5029	0.5029	0.1658	0.4195
-31	0.4718	0.4718	0.1292	0.4202
-29	0.4576	0.4576	0.0883	0.4155
-27	0.3738	0.3738	0.0532	0.4127
-25.25	0.2842	0.2842	0.0284	0.4126
-23.5	0.2121	0.2121	0.0170	0.3777
-21.25	0.1600	0.1600	0.0104	0.2781
-19	0.1177	0.1177	0.0035	0.2250
-16.75	0.0798	0.0798	0.0017	0.1956
-14.5	0.0549	0.0549	0.0010	0.1323
-13	0.0338	0.0338	0.0009	0.1213
-11.5	0.0281	0.0281	0.0007	0.0677
-9.25	0.0250	0.0250	0.0007	0.0444
-7	0.0176	0.0176	0	0.0322
-4.7143	0.0138	0.0138	0	0.0254

Dari tabel 4.9 didapatkan kurva BER pada kondisi LOS dan NLOS dengan level daya pancar yang berubah-ubah pada kondisi jarak 4 meter dan 24 meter seperti pada gambar 4.20

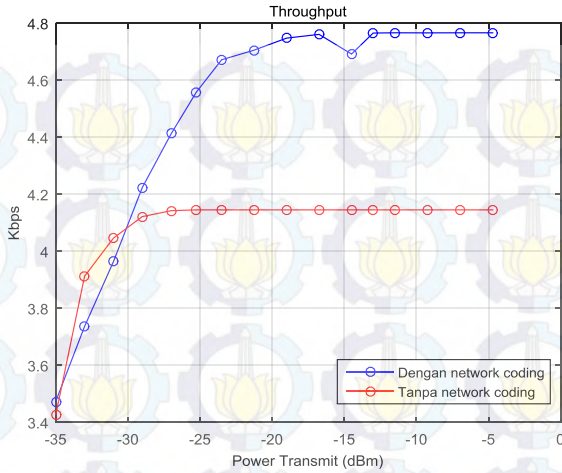


Gambar 4. 20 BER sistem terhadap fungsi jarak dan Ptx skema NC

Dari gambar 4.20 dapat dilihat bahwa untuk kondisi LOS lebih bagus dibandingkan dengan kondisi NLOS pada jarak yang jauh, sedangkan ketika jarak antara node *source* dan node *destination* berada dekat dengan node *relay* maka kondisi NLOS lebih baik kinerjanya dibandingkan dengan LOS

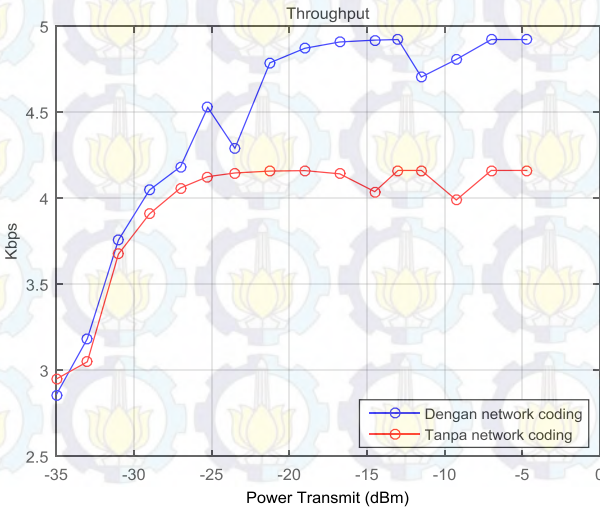
4.6 Analisis *Throughput* Hasil Pengukuran

Throughput merupakan parameter penting untuk menunjukkan tingkat performansi dari suatu jaringan dalam pengiriman paket, semakin besar nilai *Throughput* biasanya mengindikasikan bahwa kinerja dari suatu sistem akan semakin bagus. Analisis pengukuran *throughput* ini dilakukan pada kondisi lingkungan *indoor* dan *outdoor* dengan kondisi LOS dan NLOS. Berikut ini adalah gambar untuk semua kondisi:



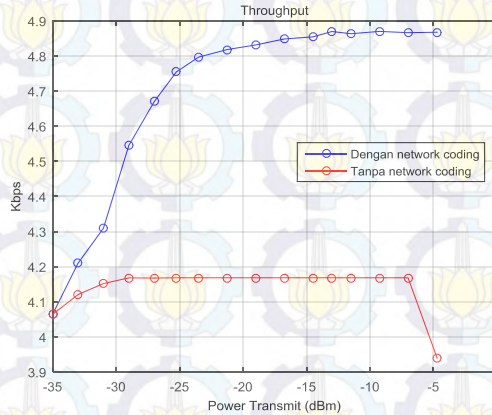
Gambar 4.21 Throughput lingkungan indoor kondisi LOS

Pada gambar 4.21 merupakan kondisi LOS ketika berada pada lingkungan indoor. Dengan persentase kenaikan throughput untuk skema network coding adalah sekitar $= \frac{4.8-4.19}{4.19} \times 100\% = 14\%$



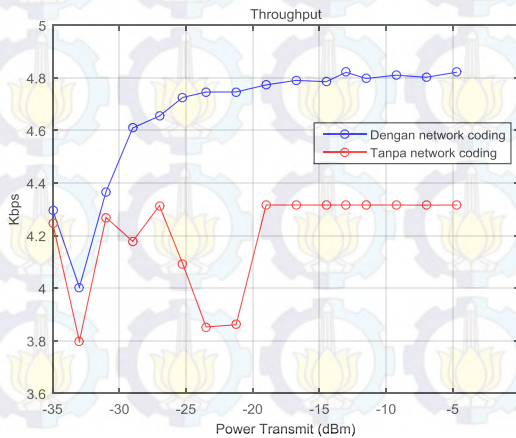
Gambar 4.22 Throughput lingkungan indoor kondisi NLOS

Pada gambar 4.22 merupakan kondisi NLOS ketika berada pada lingkungan *indoor*. Dengan persentase kenaikan *throughput* untuk skema *network coding* adalah sekitar $\frac{4.9-4.1}{4.1} \times 100\% = 20\%$



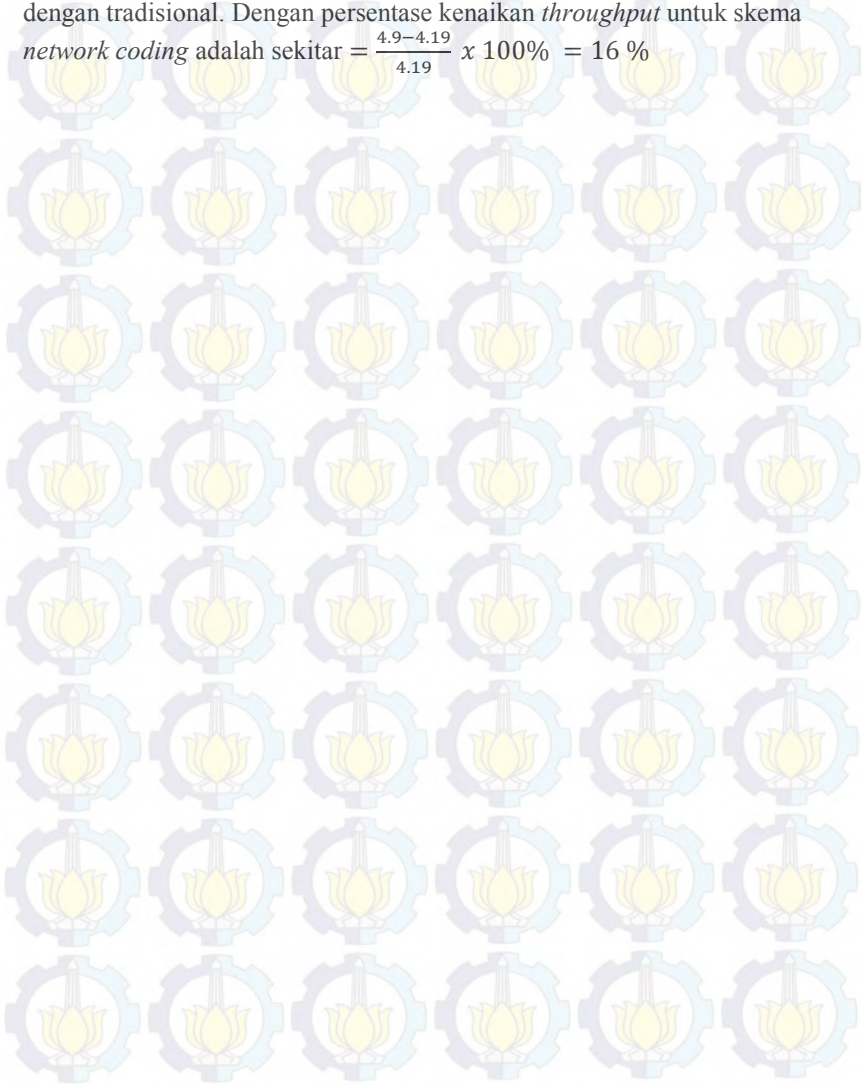
Gambar 4. 23 *Throughput* lingkungan *outdoor* kondisi LOS

Gambar 4.23 merupakan kondisi LOS ketika berada pada lingkungan *outdoor*. Dengan persentase kenaikan *throughput* untuk skema *network coding* adalah sekitar $\frac{4.9-4.19}{4.19} \times 100\% = 16\%$



Gambar 4. 24 *Throughput* lingkungan *outdoor* kondisi LOS

Berdasarkan Gambar 4.24 *throughput* diatas dapat dilihat bahwa skema *network coding* memiliki nilai *throughput* yang bagus dibandingkan dengan tradisional. Dengan persentase kenaikan *throughput* untuk skema *network coding* adalah sekitar $= \frac{4.9-4.19}{4.19} \times 100\% = 16 \%$



BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan serangkaian tinjauan pustaka, perencanaan, implementasi, pengukuran dan analisis sistem, maka dapat ditarik kesimpulan:

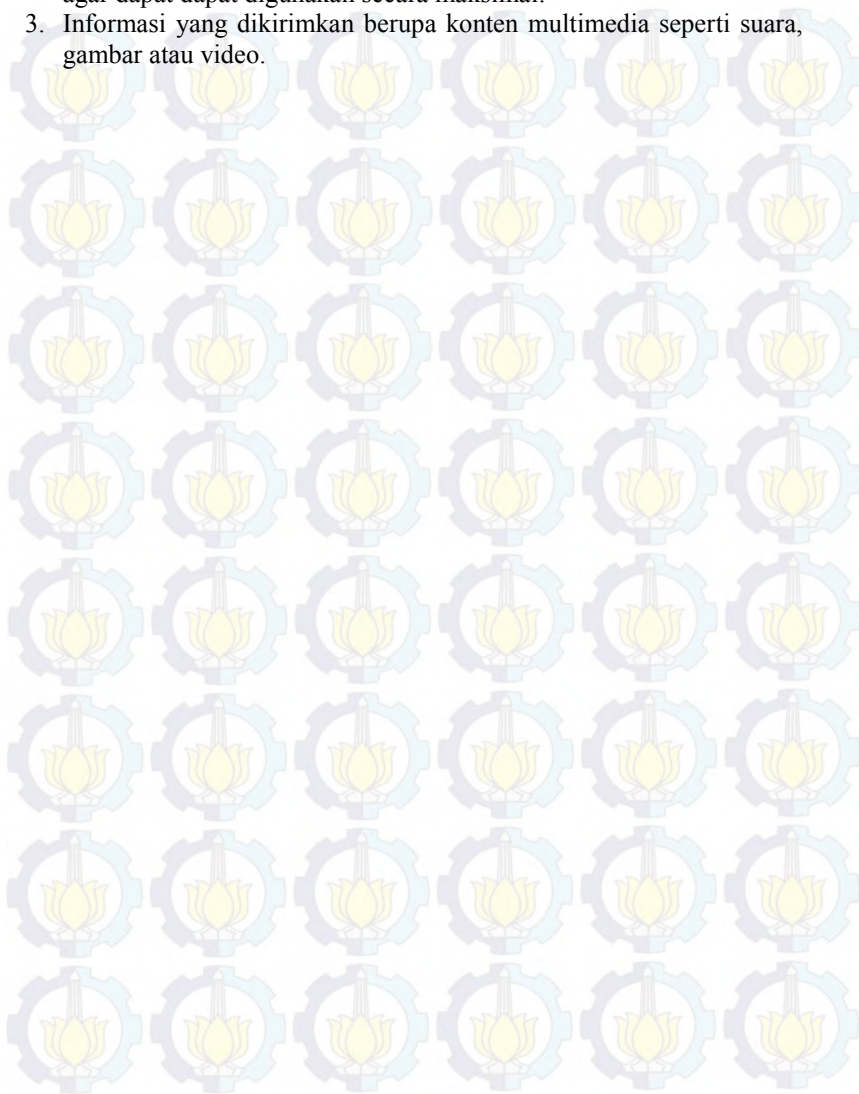
1. Penggunaan skema *physical network coding* pada sistem komunikasi kooperatif *amplify and forward* dapat meningkatkan *throughput* 20% dari skema tradisional.
2. Penggunaan skema *network coding* pada sistem komunikasi kooperatif *amplify and forward* dapat mengurangi jumlah timeslot yang dipakai dari sistem komunikasi kooperatif tradisional.
3. Untuk komunikasi multihop performansi dan kinerja BER untuk jalur multihop lebih baik dibandingkan dengan jalur SISO pada semua kondisi LOS maupun NLOS.
4. Untuk sistem komunikasi kooperatif dengan skema tradisional ditinjau dari BER, memiliki performansi yang bagus dibandingkan dengan skema *network coding*.
5. Berdasarkan fungsi jarak, BER pada sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* saat kondisi LOS pada jarak 30 meter memiliki performansi yang lebih baik dibandingkan pada jarak 4 meter.
6. Berdasarkan fungsi jarak, BER pada sistem komunikasi kooperatif *amplify and forward* dengan skema *network coding* saat kondisi NLOS pada jarak 4 meter memiliki performansi yang lebih baik dibandingkan pada jarak 30 meter.
7. Implementasi dengan menggunakan modul SDR yaitu WARP, dapat diamati performa sistem terhadap pengaruh jarak antara node *source*, *relay*, dan *destination*. Kanal yang digunakan pada modul WARP adalah kanal *real*.

5.2 Saran

Saran yang bisa diajukan sebagai bahan penelitian lebih lanjut yaitu:

1. Untuk meningkatkan kinerja BER pada sistem komunikasi kooperatif dengan skema *network coding* maka sistem perlu didesain ulang menggunakan metode jaringan *network coding* yang baru dengan menggunakan *Error Control Coding* tertentu pada sistem.

2. Lebih eksplorasi tentang *Wireless Open-Access Research Platform* agar dapat dapat digunakan secara maksimal.
3. Informasi yang dikirimkan berupa konten multimedia seperti suara, gambar atau video.



LAMPIRAN A

Jurusan Teknik Elektro - FTI
Institut Teknologi Sepuluh Nopember

TE 141599 TUGAS AKHIR - 4 SKS

Nama Mahasiswa : Ummul Khair
Nomor Pokok : 2213106010
Bidang Studi : Teknik Telekomunikasi dan Multimedia
Tugas Diberikan : Semester Gasal 2015/2016
Dosen Pembimbing : Dr. Ir. Suardi, MT
Dr. Ir. Winawan, DEA

16 SEP 2015

Judul Tugas Akhir : Implementasi dan Analisa Kinerja Sistem Komunikasi Kooperatif *Amplify and Forward* dengan skema *Network Coding* pada modul WARP (*Wireless Open-Access Research Platform*)
Implementation and Performance Analysis Of Amplify and Forward Cooperative Communication System with Network Coding Scheme using WARP (Wireless Open-Access Research Platform) module

Uraian Tugas Akhir :

Network Coding adalah teknik di mana data yang ditransmisikan dicodekan dan didekodekan untuk meningkatkan *throughput*, mengurangi keterlambatan dan membangun jaringan yang lebih kuat. *Physical Network Coding (PNC)* pada awalnya diusulkan sebagai cara untuk meningkatkan performa *network coding* menggunakan gelombang elektromagnetik dengan menambahkan 2 tingkat Komunikasi Kooperatif mentransmisikan informasi sumber dan informasi lainnya. Skema *amplify and forward (AF)* sistem komunikasi kooperatif adalah *relay* dari node pertama menerima pesan dari node sumber tanpa melalui tahap *decoding* dan *encoding*, dan langsung memperkuat dan mengirimkan ke node tujuan. Pada Tugas Akhir ini akan dilakukan simulasi kinerja dari sistem komunikasi kooperatif AF tanpa dan dengan *network coding* dan akan di implementasikan pada modul WARP (*Wireless Open-Access Research Platform*), parameter yang diambil sebagai acuan performansi adalah *bit error rate (BER)*. Diharapkan dengan ditambahkan *network coding (PNC)* pada sistem komunikasi kooperatif dengan *relay AF* kinerja sistem komunikasi kooperatif akan meningkat.

Kata Kunci : Sistem Komunikasi Kooperatif, *Amplify and Forward*, *Physical Network Coding*, WARP

Dosen Pembimbing I,

Dr. Ir. Suardi, MT
Nip. : 19680818 1993031 002

Mengesahkan,
Jurusan Teknik Elektro FTI-ITS
Kedua,

Dr. Tri Arif Sutjipto, ST, MT
nip. : 19700212 1995121 001

Dosen Pembimbing II,

Dr. Ir. Winawan, DEA
Nip. : 19651109 1989031 011

Mengesahkan,
Bidang Studi Telekomunikasi Multimedia
Koordinator,

Dr. Is Endryono, D.J.I.S
Nip. : 19650604 1991023 001



LAMPIRAN B

PENGINTEGRASIAN ANTARA PC DENGAN WARP

Langkah – langkah pengintegrasian antara PC dengan WARP

1. Menghubungkan PC dan modul WARP menggunakan kabel LAN yang terhubung pada *switch gigabit ethernet* seperti yang ditunjukkan pada gambar 3.5.
2. Mengatur IP pada PC sesuai dengan IP ketentuan perangkat WARP yaitu 10.0.0.xx dengan subnet mask 255.255.255.0



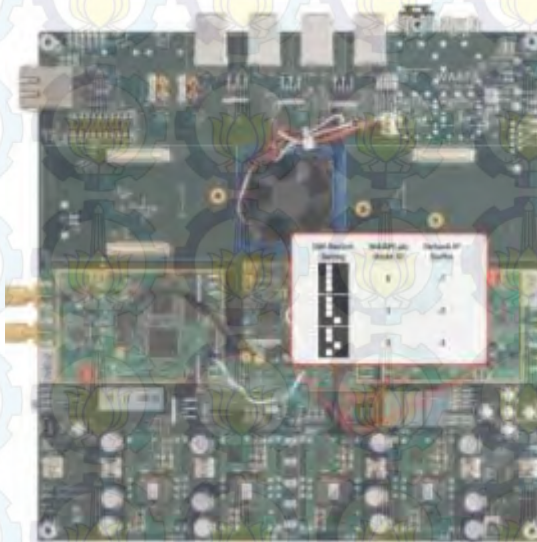
3. Mengkonfigurasi *Address Resolution Protocol* (ARP) dengan *command prompt* dan masuk sebagai administrator dengan mengetikkan *comment* :

arp -a : Mengecek tabel ARP

arp -s 10.0.0.255 ff-ff-ff-ff-ff-ff : Mengeset ARP

4. Mengatur nomor IP pada perangkat WARP menjadi 10.0.0.1 untuk node 1 (pengirim/penerima), 10.0.0.2 untuk node 2 (relay), dan 10.0.0.3 untuk node 3 (pengirim/penerima). Pengeturan dilakukan dengan cara mengatur *dipswitch* seperti gambar 3.7.
5. Memeriksa koneksi antara PC dengan perangkat WARP. Pemeriksaan koneksi antara PC dan WARP dapat dilakukan melalui *command prompt* dengan melakukan ping pada masing-masing IP node atau

melalui matlab, jika menggunakan matlab maka pada *comand window* diketikkan “wl_setup” untuk pengesetan interkoneksi antara PC dengan perangkat kemudian memasukkan IP PC yang digunakan, setelah pengesetan *wl_setup* langkah selanjutnya yaitu dengan mengetikkan “wl_initNodes(3)”. Angka (3) dari perintah *wl_initNodes(3)* mengartikan bahwa terdapat tiga buah WARP yang terkoneksi. Indikator keberhasilan koneksi dapat dilihat pada *output command window* matlab seperti yang ditunjukkan pada gambar dibawah.



```
>> wl_initNodes(3)

ans =

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 |
```

LAMPIRAN B PENGUKURAN

Pengukuran Indoor



Pengukuran Outdoor LOS





LAMPIRAN C

LISTING PROGRAM

```
%% MAIN PROGRAM
clear all;
close all;
clc;
warning off

Inisialisasi;
FrameTX;
select_mode = 0;
if (select_mode)
    fc = 2.4;
    channel = 6;
    TxGainRF = 0:4:63;
    RxGainBB = 1;
    RxGainRF = 1;
    gains = TxGainRF;
else
    EbNodB = [0:4:63];
    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));
temp_dat_set5 = zeros(stat_iter,length(gains));
temp_dat_set6 = zeros(stat_iter,length(gains));
temp_dat_set7 = zeros(stat_iter,length(gains));
temp_dat_set8 = zeros(stat_iter,length(gains));
temp_dat_set9 = zeros(stat_iter,length(gains));
temp_cyc_set1 = [];
temp_cyc_set2 = [];
% BER Container
BER_NC_S1 = zeros(size(gains));
BER_NC_S2 = zeros(size(gains));
BER_SC_S1 = zeros(size(gains));
BER_SC_S2 = zeros(size(gains));
BER_SC1Non = zeros(size(gains));
BER_SD_S1 = zeros(size(gains));
BER_SD_S2 = zeros(size(gains));
BER_MultiS1 = zeros(size(gains));
BER_S1Non = zeros(size(gains));
BER_S2Non = zeros(size(gains));
```

```

T_NCS1      = [];
T_NCS2      = [];
T_TCS1      = [];
T_TCS2      = [];
nn_iter     = 1;

% Network Coding Kooperatif
for iterasil = stat_iter
    error_nc1 = [];
    error_nc2 = [];
    error_nc3 = [];
    error_nc4 = [];
    for gain = 1 : length(gains)
        tic;
        % transmit user 1 to relay
        if (select_mode)
            [rx_vec_air_SR1,rx_vec_air_SD1,rx_RSSI_SR1,rx_RSSI_SD1,rx_ga
            ins_SR1,rx_gains_SD1] =
            warp_transmit(tx_vec_air_SD1,TxGainRF(gain),RxGainBB,RxGainR
            F,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
        % transmit user 2 to relay
        if (select_mode)
            [rx_vec_air_SR2,rx_vec_air_SD2,rx_RSSI_SR2,rx_RSSI_SD2,rx_ga
            ins_SR2,rx_gains_SD2] =
            warp_transmit1(tx_vec_air_SD2,TxGainRF(gain),RxGainBB,RxGain
            RF,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
        % RelayNC
        FrameRelay;
        % transmit relay to user 1 and user 2
        if (select_mode)
            [rx_vec_air_RS1,rx_vec_air_RD1,rx_RSSI_RS1,rx_RSSI_RD1,rx_ga
            ins_RS1,rx_gains_RD1] =

```

```

warp_transmit2(tx_vec_air_RD,TxGainRF(gain),RxGainBB,RxGainR
F,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

    FrameRX1;
    FrameRX2;
    MRC;
    SC;

    %Result
    result_BER;
    Tnc=toc;
    temp_cyc_set1(iterasil,gain) = Tnc;

    % save each statistical iteration data
    temp_dat_set1(iterasil,gain) = BER_NC_S1;
    temp_dat_set2(iterasil,gain) = BER_NC_S2;
    temp_dat_set8(iterasil,gain) = BER_SC_S1;
    temp_dat_set3(iterasil,gain) = BER_SD_S1;
    temp_dat_set4(iterasil,gain) = BER_SD_S2;

    clear sym_SD1 sym_RD1 sym_SR1 sym_SR2 sym_SD2
sym_RD2
    fprintf('Done for Network Coding = %d\n',gain);
end
    nn_iter = nn_iter+1;

    T_NCS1 = [T_NCS1 (length(s1)-error_nc1)/Tnc];
    T_NCS2 = [T_NCS2 (length(s2)-error_nc2)/Tnc];
end

time_nc_mean = mean (temp_cyc_set1,1);

% Non Network Coding Kooperatif
for iterasil = stat_iter
    error_tc1 = [];
    error_tc2 = [];
    error_tc3 = [];
    error_tc4 = [];
    for gain = 1 : length(gains)
        tic;
        % transmit user 1 to relay
        if (select_mode)

```



```

[rx_vec_air_SR1,rx_vec_air_SD1,rx_RSSI_SR1,rx_RSSI_SD1,rx_ga
ins_SR1,rx_gains_SD1] =
warp_transmit(tx_vec_air_SD1,TxGainRF(gain),RxGainBB,RxGainR
F,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
% transmit user 2 to relay
if (select_mode)

[rx_vec_air_SR2,rx_vec_air_SD2,rx_RSSI_SR2,rx_RSSI_SD2,rx_ga
ins_SR2,rx_gains_SD2] =
warp_transmit1(tx_vec_air_SD2,TxGainRF(gain),RxGainBB,RxGain
RF,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

% RelayNonNC
Relay1;
Relay2;
if select_mode
[rx_vec_air_RD1,rx_RSSI_RD1,rx_gains_RD1] =
warp_transmit1siso(tx_vec_air_RD1,TxGainRF(gain),RxGainBB,Rx
GainRF,fc,channel);
else
rx_vec_air_RD1 =
awgn(tx_vec_air_RD1,EbNodB(gain));
end

if select_mode
[rx_vec_air_RD2,rx_RSSI_RD2,rx_gains_RD2] =
warp_transmit2siso(tx_vec_air_RD2,TxGainRF(gain),RxGainBB,Rx
GainRF,fc,channel);
else
rx_vec_air_RD2 =
awgn(tx_vec_air_RD2,EbNodB(gain));
end
Frame_rx1;
Frame_rx2
MRCnon;
SCnon;

```

```

%Result
result_BERNon;
Ttc=toc;
temp_cyc_set2(iterasil,gain) = Ttc;
% save each statistical iteration data
temp_dat_set5(iterasil,gain) = BER_S1Non;
temp_dat_set6(iterasil,gain) = BER_S2Non;
temp_dat_set7(iterasil,gain) = BER_MultiS1;
temp_dat_set9(iterasil,gain) = BER_SC1Non;

clear sym_SD1 sym_RD1 sym_SR1 sym_SR2 sym_SD2
sym_RD2
fprintf('Done for Non Network Coding = %d\n',gain);
end
nn_iter = nn_iter+1;

T_TCS1 = [T_TCS1 (length(s1)-error_tc1)/Ttc];
T_TCS2 = [T_TCS2 (length(s2)-error_tc2)/Ttc];
end
time_tc_mean = mean (temp_cyc_set2,1);

% save statistical average
BER_NC_S1_mean = mean (temp_dat_set1,1);
BER_NC_S2_mean = mean (temp_dat_set2,1);
BER_SC_S1_mean = mean (temp_dat_set8,1);
BER_SISOS1_mean = mean (temp_dat_set3,1);
BER_SISOS2_mean = mean (temp_dat_set4,1);
BER_S1non_mean = mean (temp_dat_set5,1);
BER_S2non_mean = mean (temp_dat_set6,1);
BER_MULT1_mean = mean (temp_dat_set7,1);
BER_SC1Non_mean = mean (temp_dat_set9,1);
%% plot
if (select mode)
% convert gain setting to Tx RF Output in dBm according
to MAX=2828/9
tx_gain_set = [8;16;24;32;40;48;56;63];
eqv_gain = [-31;-27;-23.5;-19;-14.5;-11.5;-7;-3];
tx_outp =
interp1(tx_gain_set,eqv_gain,gains,'linear','extrap');

% plot curve
figure(1);
clf;
semilogy(tx_outp,BER_NC_S1_mean,'bo-
',tx_outp,BER_S1non_mean,'ro-',tx_outp,BER_SISOS1_mean,'go-
'); grid on;
title('Performa BER NC MRC S1, Tradisional dan Direct
Link NC');
xlabel('Tx RF output (dBm)');
ylabel('Bit Error Rate');

```

```

legend('Dengan network coding', 'Tanpa network
coding', 'Jalur S1 ke Jalur S2');

figure(2);
clf;
semilogy(tx_outp, BER_SC_S1_mean, 'bo-
', tx_outp, BER_SC1Non_mean, 'ro-', tx_outp, BER_SISOS1_mean, 'go-
'); grid on;
title('Performa BER NC SC S1, Tradisional dan Direct Link
NC');
xlabel('Tx RF output (dBm)');
ylabel('Bit Error Rate');
legend('Dengan network coding', 'Tanpa network
coding', 'Jalur S1 ke Jalur S2');

figure(3)
semilogy(tx_outp, BER_SISOS1_mean, 'bo-
', tx_outp, BER_MULTI_mean, 'ro-'); grid on;
title('Performa BER Terhadap Relay');
xlabel('Tx RF output (dBm)');
ylabel('Bit Error Rate');
legend('SISO', 'Multihop');

figure(4)
plot(tx_outp, time_nc_mean, 'bo-
', tx_outp, time_tc_mean, 'ro-'); grid on;
title('Waktu satu cycle TX/RX');
xlabel('Power Transmit (dBm)');
ylabel('Second');
legend('Network coding', 'Multihop konvensional');

figure(5)
plot(tx_outp, T_NCS1/1e3, 'bo-', tx_outp, T_TCS1/1e3, 'ro-');
grid on;
title('Throughput');
xlabel('Gain TX (dB)');
ylabel('Kbps');
legend('Dengan network coding', 'Tanpa network coding');
else
figure(1);
clf;
semilogy(EbNodB, BER_NC_S1_mean, 'bo-
', EbNodB, BER_S1non_mean, 'ro-', EbNodB, BER_SISOS1_mean, 'go-
', EbNodB, BER_SISOS2_mean, 'yo-'); grid on;
title('Performa BER NC S1, Tradisional dan Direct Link
NC');
xlabel('EbNo (dB)');
ylabel('Bit Error Rate');

```

```

legend('Dengan network coding','Tanpa network
coding','Jalur S1','Jalur S2');

```

```

figure(2);
clf;
semilogy(EbNodB,BER_NC_S2_mean,'bo-
',EbNodB,BER_S2non_mean,'ro-',EbNodB,BER_SISOS1_mean,'go-
',EbNodB,BER_SISOS2_mean,'yo-'); grid on;
title('Performa BER NC S2, Tradisional dan Direct Link
NC');
xlabel('EbNo(dB)');
ylabel('Bit Error Rate');
legend('Dengan network coding','Tanpa network
coding','Jalur S1','Jalur S2');

```

```

figure(3)
semilogy(EbNodB,BER_SISOS1_mean,'bo-
',EbNodB,BER_NC_S1_mean,'go-'); grid on;
title('Performa BER Terhadap Relay');
xlabel('EbNo(dB)');
ylabel('Bit Error Rate');
legend('SISO','Multihop');

```

```

1 figure(5)
semilogy(EbNodB,T_NCS1/1e3,'bo-',EbNodB,T_NCS2/1e3,'bo-
',EbNodB,T_TCS1/1e3,'ro-',EbNodB,T_TCS2/1e3,'go-'); grid on;
title('Throughput');
xlabel('Gain TX (dB)');
ylabel('Kbps');
legend('Dengan network coding S1','Dengan network coding
S2','Tanpa network coding S1','Tanpa network coding S2');
end

```

```

%% Simpan Pengukuran
% Time PC
format shortg;
waktu = clock;
tahun = num2str(waktu(1,1));
bulan = num2str(waktu(1,2));
hari = num2str(waktu(1,3));
jam = num2str(waktu(1,4));
menit = num2str(waktu(1,5));

```

```

simpan=input('simpan hasil pengukuran ? (1/0) :');
% Save File
if(simpan==1)
nama_file1 = 'Hasil_pengukuran';
loc = 'D:\1_Data Pengukuran Team\SemGanjil 2016\UMMUL
KHAIR\Pengukuran Indoor aj403\Result\NsymBit';
if(select_mode)

```

```

        loc_fix = strcat
        (loc, '\', 'Implementasi_', hari, bulan, tahun, '_', jam, '.', menit,
        '\');
    else
        loc_fix = strcat
        (loc, '\', 'Simulasi_', hari, bulan, tahun, '_', jam, '.', menit, '\');
    ;
end
mkdir (loc_fix);
save([loc_fix nama_file1 '.mat']);

nama_file2 = 'Grafik_BER_S1_to_Desti2';
saveas(figure(1), [loc_fix nama_file2 '.fig']);
print(figure(1), '-dbitmap', [loc_fix nama_file2 '.bmp']);

nama_file2 = 'Grafik_BER_S2_to_Desti1';
saveas(figure(2), [loc_fix nama_file2 '.fig']);
print(figure(2), '-dbitmap', [loc_fix nama_file2 '.bmp']);

nama_file2 = 'Grafik_BER_SISO_dan_Multi';
saveas(figure(3), [loc_fix nama_file2 '.fig']);
print(figure(3), '-dbitmap', [loc_fix nama_file2 '.bmp']);

nama_file2 = 'Grafik_Throughput';
saveas(figure(4), [loc_fix nama_file2 '.fig']);
print(figure(4), '-dbitmap', [loc_fix nama_file2 '.bmp']);

nama_file2 = 'Grafik_Throughput1';
saveas(figure(5), [loc_fix nama_file2 '.fig']);
print(figure(5), '-dbitmap', [loc_fix nama_file2 '.bmp']);
else
    return;
end

%% FrameTX.m
%Random binary
n_bits = 16384;

s1 = randi([0 1], n_bits, 1);
s2 = randi([0 1], n_bits, 1);
n_syms = n_bits/k;
n_sym1 = reshape(s1, n_syms, k);
n_dat1 = bi2de(n_sym1, 'left-msb');
n_sym2 = reshape(s2, n_syms, k);
n_dat2 = bi2de(n_sym2, 'left-msb');

%Hitung total data payload yang dibutuhkan,
Rsym = 5e6;
nsamp = floor(Fs/Rsym);

```

```

%Nilai nsamp harus integer, setelah pembulatan kebawah,
hitung rate simbol
%sebenarnya.
Rsym = Fs/nsamp;

%Pembentukan Preamble
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
corr_win = 300; %at least corr_win >= pream_len*nsamp
pream_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
txDelay = 400;

%Phase ambiguity detector/ Channel State
CSI_syms = 150; %Jumlah simbol untuk CSI
CSI_dat = ones(CSI_syms,1);

%Desain Square Root Raised Cosine
shape = 'Square Root Raised Cosine';
N_Ord = 6; %Genap
R_SRRC = 0.9; %[0-1]
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
data_sym = floor((L-(pream_len*nsamp)-txDelay-
flt_delay)/nsamp);

%Baseband Modulation
sym11 = modulate(h_bb_mod,n_dat1);
sym22 = modulate(h_bb_mod,n_dat2);

%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 = [sym11;zeros(pad,1)];
sym2 = [sym22;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

sym111 = sym1(1:length(n_dat1));
sym222 = sym2(1:length(n_dat2));
%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_fra
mes)]);
xt_mf1 = zeros(size(xt_mf_bb1));

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

%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
%% Inisialisasi m

%Frek. sampling dan waktu sampling
Fs = 40e6;
Ts = 1/Fs;

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

%Sistem modulasi yang digunakan
M = 4;
k = log2(M);
if M > 2
    ph_off = 0;
    h_bb_mod =
modem.pskmod('M',M,'PhaseOffset',ph_off,'SymbolOrder','Gray'
);
    h_bb_demod =
modem.pskdemod('M',M,'PhaseOffset',ph_off,'SymbolOrder','Gra
y');
else
    h_bb_mod = modem.pskmod('M',M);
    h_bb_demod = modem.pskdemod('M',M);
end

function
[rx_vec_air_SR1,rx_vec_air_SD1,rx_RSSI_SR1,rx_RSSI_SD1,rx ga
ins_SR1,rx_gains_SD1] =
warp_transmit(tx_vec_air_SD1,TxGainRF,RxGainBB,RxGainRF,fc,c
hannel)
% Detailed explanation goes here
NUMNODES = 3;

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

```



```

fprintf('Begin WARP Tx-Rx...\n')
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]
);

[RFA,RFB] = wl_getInterfaceIDs(nodes(1));

%Set up the interface for the experiment
wl_interfaceCmd(nodes,'RF_ALL','tx_gains',2,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');
wl_interfaceCmd(nodes,'RF_ALL','rx_gains',RxGainRF,RxGainBB)
;
end

txLength = nodes(1).baseband.txIQLen;

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

node_tx = nodes(1);
node_rx_1 = nodes(2);
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'));

tx_vec_air_SD1_size = size(tx_vec_air_SD1);
rx_vec_air_SR1 = zeros(txLength,tx_vec_air_SD1_size(2));
rx_vec_air_SD1 = zeros(txLength,tx_vec_air_SD1_size(2));
rx_RSSI = zeros(txLength/4,tx_vec_air_SD1_size(2));
rx_gains = zeros(3,tx_vec_air_SD1_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_SD1_size(2);
    wl_basebandCmd(node_tx,[RF_TX], 'write_IQ',
tx_vec_air_SD1(:,m) );
    %send trigger
    eth_trig.send();
    %Get received data

    %First RF
    rx_vec_air_SR1(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'read_IQ', 0, txLength);
    rx_vec_air_SD1(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX], 'read_IQ', 0, txLength);

    rx_RSSI_SR1(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'read_RSSI',0,txLength/(Ts_
RSSI/Ts));
    rx_RSSI_SD1(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX], 'read_RSSI',0,txLength/(Ts_
RSSI/Ts));

    rx_gains_SR1(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'agc_state');
    rx_gains_SD1(:,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_SR2,rx_vec_air_SD2,rx_RSSI_SR2,rx_RSSI_SD2,rx ga
ins_SR2,rx_gains_SD2] =
warp_transmit1(tx_vec_air_SD2,TxGainRF,RxGainBB,RxGainRF,fc,
channel)

% Detailed explanation goes here
NUMNODES = 3;

% Create a vector of node objects

```

```

nodes      = wl_initNodes(NUMNODES);

fprintf('Begin WARP Tx-Rx...\n')
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]
);
[RFA,RFB]  = wl_getInterfaceIDs(nodes(3));

%Set up the interface for the experiment
wl_interfaceCmd(nodes,'RF_ALL','tx_gains',2,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');

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

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'));

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_SR2(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'read_RSSI',0,txLength/(Ts_
RSSI/Ts));
    rx_RSSI_SD2(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX], 'read_RSSI',0,txLength/(Ts_
RSSI/Ts));

    rx_gains_SR2(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'agc_state');
    rx_gains_SD2(:,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

%% FrameRelay.m
time1 = ((0:length(rx_vec_air_SR1)-1)/Fs).';
rx_IQ_disp1 = rx_vec_air_SR1;

time2 = ((0:length(rx_vec_air_SR2)-1)/Fs).';
rx_IQ_disp2 = rx_vec_air_SR2;

for ii=1:n_frames
    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

%Sinkronisasi fasa
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));
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);

    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));
%   temp_vct32 = yt_rx_data2(:,ii).*conj(ch_est2(:,ii));
multip1 = [abs(ch_est1(:,ii)).^2];
multip2 = [abs(ch_est2(:,ii)).^2];
%   multip = sum(multip,2);
yt_rx_data1(:,ii) = (r_sym1)./multip1;
yt_rx_data2(:,ii) = (r_sym2)./multip2;
%   a = r_sym1;
%   b = a./conj(ch_est(:,ii));

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

z_sym1(:,ii) = sym_stuf1;
z_sym2(:,ii) = sym_stuf2;
end

%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_SR1 = reshape(z_sym1,data_sym*n_frames,1);
sym_SR1 = sym_SR1(1:length(n_dat1)); %% data yang
diterima di relay

sym_SR2 = reshape(z_sym2,data_sym*n_frames,1);
sym_SR2 = sym_SR2(1:length(n_dat2)); %% data yang
diterima di relay

symSR = sym_SR1 + sym_SR2;

% Demodulasi
z_symR = demodulate(h_bb_demod,symSR);

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

yy1= de2bi(rx_relay,'left-msb');
rx_bitR=reshape(yy1,1,[]);

```

```

bit_SR = rx_bitR';

%% Modulator in relay
n_symSR = reshape(bit_SR,n_syms,k);
n_datSR = bi2de(n_symSR,'left-msb');
symSR = modulate(h_bb_mod,n_datSR); %s1+s2

%Tambahkan zero padding pada symbol
symSR = [symSR;zeros(pad,1)];
%Reshape n_sym_dat
symSR = reshape(symSR,data_sym,n_frames); % Sebelum dikirim
ke WARP, yg dikirim

%Bagi data menjadi beberapa frame, tambahkan bit stuffing
for ii=1:n_frames
    n_dat_fr(:,ii) =
sym_stuffing(n_dat_fr(:,ii),symSR(:,ii),sym_size);
end

%Tambahkan preamble
xt_dat = [temp_mat1;n_dat_fr];
xt_mf_bb =
filter(h_SRRC,[upsample(xt_dat,nsamp);zeros(flt_delay,n_frames)
]);
xt_mf = zeros(size(xt_mf_bb));

%Upconvert semua frame, normalisasi
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

function
[rx_vec_air_RS1,rx_vec_air_RD1,rx_RSSI_RS1,rx_RSSI_RD1,rx_ga
ins_RS1,rx_gains_RD1] =
warp_transmit2(tx_vec_air_RD,TxGainRF,RxGainBB,RxGainRF,fc,c
hannel)

NUMNODES = 3;

% Create a vector of node objects

```

```

nodes        = wl_initNodes(NUMNODES);

fprintf('Begin WARP Tx-Rx...\n')
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]
);
%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', 3, 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');

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

%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(2), 'tx_buff_clk_freq'));
Ts_RSSI = 1/(wl_basebandCmd(nodes(2), 'rx_rssi_clk_freq'));

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_RS1(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'read_RSSI',0,txLength/(Ts_
RSSI/Ts));
    rx_RSSI_RD1(:,m) =
wl_basebandCmd(node_rx_2,[RF_RX], 'read_RSSI',0,txLength/(Ts_
RSSI/Ts));

    rx_gains_RS1(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX], 'agc_state');
    rx_gains_RD1(:,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

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

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

yt_rx_datas1 = zeros(size(n_dat_fr1));
yt_rx_datas2 = zeros(size(n_dat_fr2));

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

ch_est1 = zeros(size(n_dat_fr1));
ch_est2 = zeros(size(n_dat_fr2));
%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,[]);
bit_SD1 = rx_bit1';

yy2= de2bi(rx_siso2,'left-msb');
rx_bit2=reshape(yy2,1,[]);
bit_SD2 = rx_bit2';
%%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(SSRC_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));

    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)./multipl2;

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));

%% NC Dan maximum_ratio_combining
%Data yang di proses di Relay Network coding nya

%Untuk s1
z1 = sym_RD1 - sym_SD2; %s1
%Untuk s2
z2 = sym_RS1 - sym_SD1; %s2

% MRC1
SD1 = abs(sym_SD1).*exp(1i*angle(sym_SD1));
RD1 = abs(z1).*exp(1i*angle(z1));
akhir_s1 = SD1 + RD1;

% MRC2
SD2 = abs(sym_SD2).*exp(1i*angle(sym_SD2));
RD2 = abs(z2).*exp(1i*angle(z2));
akhir_s2 = SD2 + RD2;

%% demodulator with MRC

mrc_s1 = demodulate(h_bb_demod,akhir_s1);
sym_MRC_s1 = mrc_s1(1:length(n_dat2));

bit_MRC_s1 = de2bi(sym_MRC_s1,'left-msb');
bit_MRC_s1 = reshape(bit_MRC_s1 , [],1);

mrc_s2 = demodulate(h_bb_demod,akhir_s2);
sym_MRC_s2 = mrc_s2(1:length(n_dat1));

```

```

bit_MRC_s2 = de2bi(sym_MRC_s2,'left-msb');
bit_MRC_s2 = reshape(bit_MRC_s2 ,[],1);

%% Counting BER

[N1,B1] = biterr(s1,bit_MRC_s1);
BER_NC_S1 = B1;

[N2,B2] = biterr(s2,bit_MRC_s2);
BER_NC_S2 = B2;

[N3,B3] = biterr(s1,bit_SD1);
BER_SD_S1 = B3;

[N4,B4] = biterr(s2,bit_SD2);
BER_SD_S2 = B4;

[N8,B8] = biterr(s1,bit_SC1);
BER_SC_S1 = B8;

[N9,B9] = biterr(s2,bit_SC2);
BER_SC_S2 = B9;

error_nc1 = [error_nc1 biterr(s1,bit_MRC_s1)];
error_nc2 = [error_nc2 biterr(s2,bit_MRC_s2)];

error_nc3 = [error_nc3 biterr(s1,bit_SC1)];
error_nc4 = [error_nc4 biterr(s2,bit_SC2)];

```

Tradisional

warp_transmit

warrp_transmit1

```

%% FrameRelay.m
%Sinkronisasi dan Downconvert semua frame
time1 = (0:length(rx_vec_air_SR1)-1)/Fs.';
rx_IQ_displ = rx_vec_air_SR1;

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);
end

%Sinkronisasi fasa

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_rx_data = zeros(size(n_dat_fr1));
z_syml = zeros(size(syml));
ch_est = zeros(size(n_dat_fr1));

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

    %Downsampling
    temp_vct11 = temp_vct1(first_sample1:end);

    temp_vct111 = downsample(temp_vct11(1:end),nsamp);
    %Trim
    temp_vct1111 = temp_vct111(1:fr_len);

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

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

    %Received symbols:
    r_syml = yt_rx_data1(:,ii).*conj(ch_est1(:,ii));

    multipl1 = [abs(ch_est1(:,ii)).^2];

    %    multipl = sum(multipl,2);
    yt_rx_data1(:,ii) = (r_syml)./multipl;

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

    %Potong semua frame CSI, kumpulkan hasil data pada
    matriks tertentu
    z_syml(:,ii) = sym_stuf1;

end

```



```

%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_SR1 = reshape(z_sym1,data_sym*n_frames,1);
sym_SR1 = sym_SR1(1:length(n_dat1));      %% data yang
diterima di relay

%Demodulasi
z_sym1 = demodulate(h_bb_demod,sym_SR1);

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

yy1= de2bi(rx_sisol,'left-msb');
rx_bit1=reshape(yy1,1,[]);
bit_SR1 = rx_bit1';

%% Modulatore in relay

n_symSR1 = reshape(bit_SR1,n_syms,k);
n_datSR1 = bi2de(n_symSR1,'left-msb');

symSR1 = modulate(h_bb_mod,n_datSR1);
%Tambahkan zero padding pada symbol
symSR1 = [symSR1;zeros(pad,1)];
%Reshape n_sym_dat
symSR1 = reshape(symSR1,data_sym,n_frames); % Sebelum
dikirim ke WARP, yg dikirim

%Bagi data menjadi beberapa frame, tambahkan bit stuffing
for ii=1:n_frames
    n_dat_fr(:,ii) =
sym_stuffing(n_dat_fr(:,ii),symSR1(:,ii),sym_size);
end

%Tambahkan preamble
xt_dat1 = [temp_mat1;n_dat_fr];

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

%Upconvert semua frame, normalisasi
time = [(0:length(xt_mf1)-1)/Fs].';
f = 5e6;
for ii=1:n_frames
    xt_mf1(:,ii) = xt_mf_bb1(:,ii).*exp(1i*2*pi*f*time);

```

```

    norm_fact = max( [ max(real(xt_mf1(:,ii))) ,
max(imag(xt_mf1(:,ii))) ] );
    %Normalize frame
    tx_vec_air_RD1(:,ii) = xt_mf1(:,ii)./norm_fact; %% yang
dikirim ke WARP
end

%% FrameRelay.m
%Sinkronisasi dan Downconvert semua frame
time2 = ((0:length(rx_vec_air_SR2)-1)/Fs).';
rx_IQ_disp2 = rx_vec_air_SR2;

for ii=1:n_frames
    %vco_sig = syn_frame(rx_IQ(:,ii),f,Fs);
    vco_sig2 = exp(1i*2*pi*f*time2);
    rx_vec_air_SR2(:,ii) =
rx_vec_air_SR2(:,ii).*conj(vco_sig2);
end

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

%Matched filtering
yt_mf2 =
filter(hrx_SRRC,[rx_vec_air_SR2;zeros(flt_delay,n_frames)]);
yt_rx_data = zeros(size(n_dat_fr));
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_vct2 = yt_mf2(:,ii);

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

    %Downsampling
    temp_vct22 = temp_vct2(first_sample2:end);
    temp_vct222 = downsample(temp_vct22(1:end),nsamp);
    %Trim
    temp_vct2222 = temp_vct222(1:fr_len);

```

```

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

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

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

multip2 = [abs(ch_est2(:,ii)).^2];
%   multip = sum(multip,2);
yt_rx_data2(:,ii) = (r_sym2)./multip2;

%Symbol de-stuffing
sym_stuf2 = sym_destuffing(yt_rx_data2(:,ii),sym_size);
z_sym2(:,ii) = sym_stuf2;
end

%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_SR2 = reshape(z_sym2,data_sym*n_frames,1);
sym_SR2 = sym_SR2(1:length(n_dat2));      %% data yang
diterima di relay

%Demodulasi
z_sym2 = demodulate(h_bb_demod,sym_SR2);

%Potong z_sym agar memiliki panjang sama dengan n_dat
z_sym2 = z_sym2(1:length(n_dat2));
rx_asiso2= z_sym2;

yy2= de2bi(rx_asiso2,'left-msb');
rx_bit2=reshape(yy2,1,[]);
bit_SR2 = rx_bit2';

%% Modulatore in relay
n_symSR2 = reshape(bit_SR2,n_syms,k);
n_datSR2 = bi2de(n_symSR2,'left-msb');

symSR2 = modulate(h_bb_mod,n_datSR2);
%Tambahkan zero padding pada symbol
symSR2 = [symSR2;zeros(pad,1)];
%Reshape n_sym_dat
symSR2 = reshape(symSR2,data_sym,n_frames); % Sebelum
dikirim ke WARP, yg dikirim

%Bagi data menjadi beberapa frame, tambahkan bit stuffing

```

```

for ii=1:n_frames
    n_dat_fr(:,ii) =
sym_stuffing(n_dat_fr(:,ii),symSR2(:,ii),sym_size);
end

%Tambahkan preamble
xt_dat2 = [temp_mat1;n_dat_fr];

%Interpolasi, matched filtering, tambahkan filter delay
xt_mf_bb2 =
filter(h_SRRC,[upsample(xt_dat2,nsamp);zeros(flt_delay,n_fra
mes)]);
xt_mf2 = zeros(size(xt_mf_bb2));

%Upconvert semua frame, normalisasi
time3 = [(0:length(xt_mf2)-1)/Fs].';
f = 5e6;
for ii=1:n_frames
    xt_mf2(:,ii) = xt_mf_bb2(:,ii).*exp(1i*2*pi*f*time3);
    norm_fact2 = max( [ max(real(xt_mf2(:,ii))) ,
max(imag(xt_mf2(:,ii))) ] );
    %Normalize frame
    tx_vec_air_RD2(:,ii) = xt_mf2(:,ii)./norm_fact2; %% yang
dikirim ke WARP
end

function [rx_vec_air_RD1,rx_RSSI_RD1,rx_gains_RD1] =
warp_transmit1siso(tx_vec_air_RD1,TxGainRF,RxGainBB,RxGainRF
,fc,channel)
NUMNODES = 3;
nodes = wl_initNodes(NUMNODES);

fprintf('Begin WARP Tx-Rx...\n')
USE_AGC = false;

eth_trig = wl_trigger_eth_udp_broadcast;
wl_triggerManagerCmd(nodes,'add_ethernet_trigger',[eth_trig]
);
[RFA,RFB] = wl_getInterfaceIDs(nodes(2));

%Set up the interface for the experiment
wl_interfaceCmd(nodes,'RF_ALL','tx_gains',3,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');
    wl_interfaceCmd(nodes, 'RF_ALL', 'rx_gains', RxGainRF, RxGainBB)
    ;
end
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(2), 'tx_buff_clk_freq'));
Ts_RSSI = 1/(wl_basebandCmd(nodes(2), 'rx_rssi_clk_freq'));

tx_vec_air_RD_size = size(tx_vec_air_RD1);
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_2, RF_RX, 'rx_en');

wl_basebandCmd(node_tx, RF_TX, 'tx_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_RD1(:,m) );
    %send trigger
    eth_trig.send();
    %Get received data

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

    rx_RSSI_RD1(:,m) =
    wl_basebandCmd(node_rx_2, [RF_RX], 'read_RSSI', 0, txLength/(Ts_
    RSSI/Ts));

    rx_gains_RD1(:,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_RD2,rx_RSSI_RD2,rx_gains_RD2] =
warp_transmit2siso(tx_vec_air_RD2,TxGainRF,RxGainBB,RxGainRF
,fc,channel)
% % Detailed explanation goes here
NUMNODES = 3;

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

fprintf('Begin WARP Tx-Rx...\n')
USE_AGC = false;

eth_trig = wl_trigger_eth_udp_broadcast;
wl_triggerManagerCmd(nodes,'add_ethernet_trigger',[eth_trig
]);
[RFA,RFB] = wl_getInterfaceIDs(nodes(2));

%Set up the interface for the experiment
wl_interfaceCmd(nodes,'RF_ALL','tx_gains',3,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');
    wl_interfaceCmd(nodes,'RF_ALL','rx_gains',RxGainRF,RxGainBB)
;
end

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);

RF_TX = RFA;
RF_RX = RFA;

```

```

Ts = 1/(wl_basebandCmd(nodes(2), 'tx_buff_clk_freq'));
Ts_RSSI = 1/(wl_basebandCmd(nodes(2), 'rx_rssi_clk_freq'));

tx_vec_air_RD_size = size(tx_vec_air_RD2);
rx_vec_air_RD2 = 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_basebandCmd(node_tx,RF_TX,'tx_buff_en');
wl_basebandCmd(node_rx_1,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_RD2(:,m) );
    %send trigger
    eth_trig.send();
    %Get received data

    %First RF
    rx_vec_air_RD2(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX],'read_IQ', 0, txLength);

    rx_RSSI_RD2(:,m) =
wl_basebandCmd(node_rx_1,[RF_RX],'read_RSSI',0,txLength/(Ts_
RSSI/Ts));

    rx_gains_RD2(:,m) =
wl_basebandCmd(node_rx_1,[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

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

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

```

```

    rx_vec_air_SD1(:,ii) =
rx_vec_air_SD1(:,ii).*conj(vco_sig1);
end

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

%Matched filtering
yt_mfl =
filter(hrx_SRRC,[rx_vec_air_SD1;zeros(flt_delay,n_frames)]);
yt_rx_data = zeros(size(n_dat_fr));
z_sym1 = zeros(size(sym1));
ch_est = zeros(size(n_dat_fr));

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

    %Downsampling
    temp_vct11 = temp_vct1(first_sample1:end);

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

    %Trim
    temp_vct1111 = temp_vct111(1:fr_len);

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

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

    %Received symbols:
    r_sym1 = yt_rx_data1(:,ii).*conj(ch_est1(:,ii));
    multipl = [abs(ch_est1(:,ii)).^2];

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

```



```

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

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

%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_SD1 = reshape(z_syml,data_sym*n_frames,1);
sym_SD1 = sym_SD1(1:length(n_dat1)); %% data yang diterima
di destination1

%Demodulasi
z_syml = demodulate(h_bb_demod,sym_SD1);

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

yy1= de2bi(rx_siso1,'left-msb');
rx_bit1=reshape(yy1,1,[]);
bit_SD1 = rx_bit1';

%% Jalur SISO

%Sinkronisasi dan Downconvert semua frame
time2 = ((0:length(rx_vec_air_RD1)-1)/Fs).';
rx_vec_air_RD1 = rx_vec_air_RD1;

for ii=1:n_frames
    vco_sig4 = exp(1i*2*pi*f*time2);
    rx_vec_air_RD1(:,ii) =
rx_vec_air_RD1(:,ii).*conj(vco_sig4);
end

%Sinkronisasi fasa
%Matched filtering
yt_mf2 =
filter(hrx_SRRRC,[rx_vec_air_RD1;zeros(flt_delay,n_frames)]);

z_sym2 = zeros(size(symsR1));

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

    %Cari korelasi tertinggi

```

```

correlation2 = abs( (temp_vct2(1:corr_win)') * ref_matx
);
pream_start2 = find(correlation2 == max(correlation2));
first_sample2 = pream_start2+length(pream_up);
%Downsampling
temp_vct22 = temp_vct2(first_sample2:end);
%Start sampling at initial point
temp_vct222 = downsample(temp_vct22(1:end),nsamp);
%Trim
temp_vct2222 = temp_vct222(1:fr_len);
%Save trimmed vector
yt_rx_data2(:,ii) = temp_vct2222;
%Estimasi kanal
ch_est2(:,ii) =
estimate_chan(yt_rx_data2(:,ii),sym_size);
%Received symbols:
r_sym2 = yt_rx_data2(:,ii).*conj(ch_est2(:,ii));
multip2 = [abs(ch_est2(:,ii)).^2];
yt_rx_data2(:,ii) = (r_sym2)./multip2;
%Symbol de-stuffing
sym_stuf2 = sym_destuffing(yt_rx_data2(:,ii),sym_size);
%Potong semua frame CSI, kumpulkan hasil data pada
matriks tertentu
z_sym2(:,ii) = sym_stuf2;
end
%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_RD1 = reshape(z_sym2,data_sym*n_frames,1);
sym_RD1 = sym_RD1(1:length(n_datSR1)); %% data yang
diterima di destination2
%Demodulasi
z_sym2 = demodulate(h_bb_demod,sym_RD1);
%Potong z_sym agar memiliki panjang sama dengan n_dat
z_sym2 = z_sym2(1:length(n_datSR1));
rx_siso2= z_sym2;

yy2= de2bi(rx_siso2,'left-msb');

```

```

rx_bit2=reshape(yy2,1,[]);
bit_RD1 = rx_bit2';

%% FrameRX.m
%% Jalur SISO
%Sinkronisasi dan Downconvert semua frame
time1 = ((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_SD2(:,ii) =
rx_vec_air_SD2(:,ii).*conj(vco_sig1);
end

%Sinkronisasi fasa

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

%Matched filtering
yt_mf2 =
filter(hrx_SRRC,[rx_vec_air_SD2;zeros(flt_delay,n_frames)]);
yt_rx_data = zeros(size(n_dat_fr));
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_vct2 = yt_mf2(:,ii);

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

    %Downsampling
    temp_vct22 = temp_vct2(first_sample2:end);

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

    %Trim
    temp_vct2222 = temp_vct222(1:fr_len);

    %Save trimmed vector

```

```

yt_rx_data2(:,ii) = temp_vct2222;

%Estimasi kanal
ch_est2(:,ii) =
estimate_chan(yt_rx_data2(:,ii),sym_size);
%Received symbols:
r_sym2 = yt_rx_data2(:,ii).*conj(ch_est2(:,ii));
multip2 = [abs(ch_est2(:,ii)).^2];
yt_rx_data2(:,ii) = (r_sym2)./multip2;
%Symbol de-stuffing
sym_stuf2 = sym_destuffing(yt_rx_data2(:,ii),sym_size);
%Potong semua frame CSI, kumpulkan hasil data pada
matriks tertentu
z_sym2(:,ii) = sym_stuf2;
end
%Ubah yt_rx_data menjadi vektor, potong yt_syms
sym_SD2 = reshape(z_sym2,data_sym*n_frames,1);
sym_SD2 = sym_SD2(1:length(n_dat2)); %% data yang diterima
di destination1
%Demodulasi
z_sym2 = demodulate(h_bb_demod,sym_SD2);
%Potong z_sym agar memiliki panjang sama dengan n_dat
z_sym2 = z_sym2(1:length(n_dat2));
rx_asiso2= z_sym2;
yy2= de2bi(rx_asiso2,'left-msb');
rx_bit2=reshape(yy2,1,[]);
bit_SD2 = rx_bit2';
%% Jalur SISO
%Sinkronisasi dan Downconvert semua frame
time2 = ((0:length(rx_vec_air_RD2)-1)/Fs).';
rx_vec_air_RD2 = rx_vec_air_RD2;
for ii=1:n_frames
vco_sig2 = exp(1i*2*pi*f*time2);
rx_vec_air_RD2(:,ii) =
rx_vec_air_RD2(:,ii).*conj(vco_sig2);
end

```

```

%Sinkronisasi fasa
%Matched filtering
yt_mf2 =
filter(hrx_SRRRC,[rx_vec_air_RD2;zeros(flt_delay,n_frames)]);
z_sym2 = zeros(size(symSR2));

%Deteksi simbol pertama pada tiap-tiap frame
for ii=1:n_frames
temp_vct2 = yt_mf2(:,ii);
%Cari korelasi tertinggi
correlation2 = abs( (temp_vct2(1:corr_win)') * ref_matx
);
pream_start2 = find(correlation2 == max(correlation2));
first_sample2 = pream_start2+length(pream_up);

%Downsampling
temp_vct22 = temp_vct2(first_sample2:end);

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

%Trim
temp_vct2222 = temp_vct222(1:fr_len);

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

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

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

multip2 = [abs(ch_est2(:,ii)).^2];
yt_rx_data2(:,ii) = (r_sym2)./multip2;

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

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

%Ubah yt_rx_data menjadi vektor, potong yt_syms

```

```

sym_RD2 = reshape(z_sym2,data_sym*n_frames,1);
sym_RD2 = sym_RD2(1:length(n_datSR2)); %% data yang
diterima di destination2

%Demodulasi
z_sym2 = demodulate(h_bb_demod,sym_RD2);

%Potong z_sym agar memiliki panjang sama dengan n_dat

z_sym2 = z_sym2(1:length(n_datSR2));
rx_siso2= z_sym2;

yy2= de2bi(rx_siso2,'left-msb');
rx_bit2=reshape(yy2,1,[]);
bit_RD2 = rx_bit2';

%% maximum_ratio_combining 1

SDNon1 = abs(sym_SD1).*exp(1i*angle(sym_SD1));
RDNon1 = abs(sym_RD1).*exp(1i*angle(sym_RD1));
MRCNon1 = SDNon1 + RDNon1;

% demodulator with MRC

mrc_non1 = demodulate(h_bb_demod,MRCNon1);
sym_MRCnon1 = mrc_non1(1:length(n_dat1));

bit_MRCnon1 = de2bi(sym_MRCnon1,'left-msb');
bit_MRCnon1 =reshape(bit_MRCnon1 ,[],1);

%% maximum_ratio_combining 2

SDNon2 = abs(sym_SD2).*exp(1i*angle(sym_SD2));
RDNon2 = abs(sym_RD2).*exp(1i*angle(sym_RD2));
MRCNon2 = SDNon2 + RDNon2;

% demodulator with MRC

mrc_non2 = demodulate(h_bb_demod,MRCNon2);
sym_MRCnon2 = mrc_non2(1:length(n_dat1));

bit_MRCnon2 = de2bi(sym_MRCnon2,'left-msb');
bit_MRCnon2 =reshape(bit_MRCnon2 ,[],1);

%% Counting BER

[X1,X1] = biterr(s1,bit_MRCnon1);
BER_S1Non = X1;

error_tc1 = [error_tc1 biterr(s1,bit_MRCnon1)];

```

```
[X2,X2]      = biterr(s2,bit_MRCnon2);
BER_S2Non   = X2;
error_tc2 = [error_tc2 biterr(s2,bit_MRCnon2)];

[X3,X3]      = biterr(s1,bit_RD1);
BER_MultiS1 = X3;

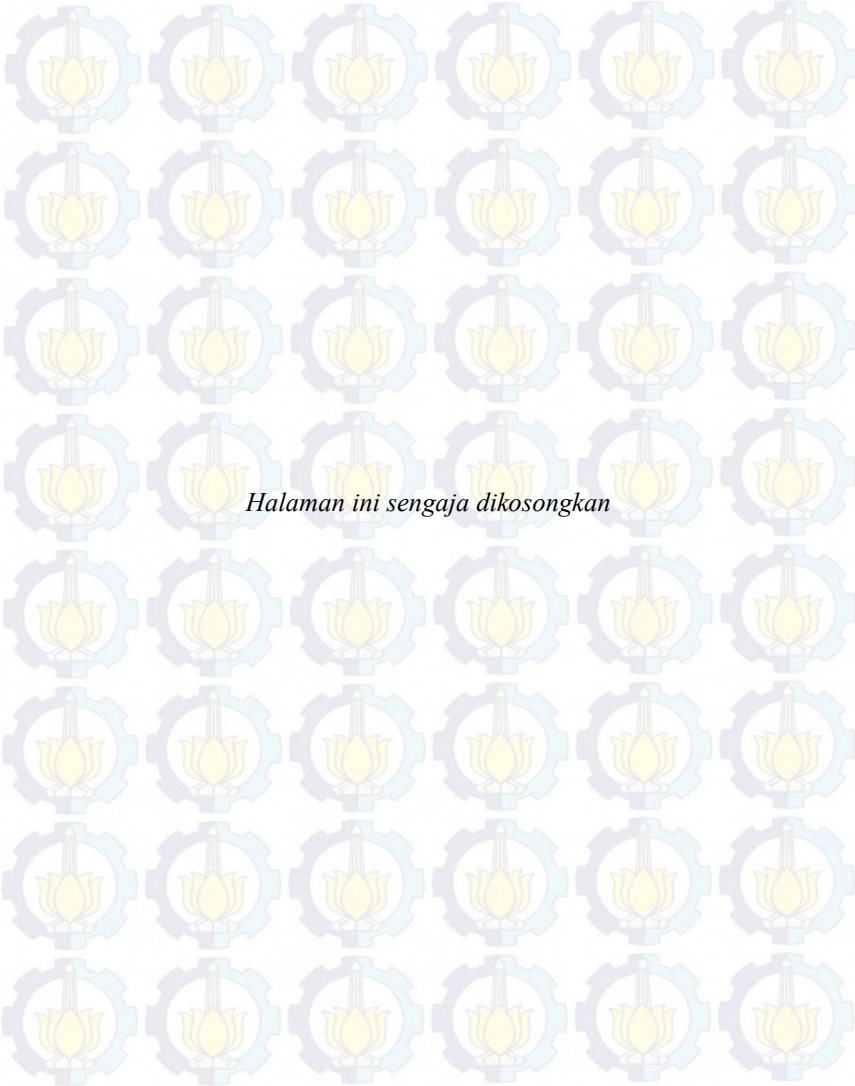
[X4,X4]      = biterr(s2,bit_RD2);
BER_MultiS2 = X4;
%%
[X5,X5]      = biterr(s1,bit_SC3);
BER_SC1Non  = X5;
error_tc3 = [error_tc3 biterr(s1,bit_SC3)];

[X6,X6]      = biterr(s2,bit_SC4);
BER_SC2Non  = X6;

error_tc4 = [error_tc4 biterr(s2,bit_SC4)]
```


DAFTAR PUSTAKA

- [1]. Rappaport, Theodore S (2002). “*Wireless Communications Principles and Practice*” Second edition. Prentice Hall, Inc, Upper Sadley River. New Jersey.
- [2]. N. Aria, T. E. Hunter, and A. Hedayat, “Cooperative Communication in Wireless Network,” Oct-2004.
- [3]. Ray Liu, K. J., K. Sadek, Ahmad., Su, Weifeng., and Kwasinski, Andres., 2009. Cooperative Communication And Networking. Cambridge University Press.
- [4]. Fragouli, Christina., and Soljanin, Emina., 2007. Network Coding Fundamentals, Foundation and Trends R in Networking, vol 2, no 1.
- [5]. Soung Chang Liew, Shengli Zhang, “Physical Layer Network Coding: Tutorial, Survey And Beyond”.,2013., Physical Communication 6 4-42.
- [6]. <https://warpproject.org> (Online), (diakses 08 Juni 2015, Jam 13.12)
- [7]. Xiaoming Liu, et al, “A Physical-Network-Coding-based Amplify and Forward Cooperation Scheme”., Harbin. IEEE, December 2013
- [8]. W Peter Hong, Wan Jen Huang, C.C Jay Kuo, “ Cooperative Communication And Networking”, Technology And System Design,2010.
- [9]. Proakis, John G., MasoudSalehi., “*Digital Communications*”, The McGraw-Hill Companies, Inc., USA, 2008.
- [10]. Zhanji, Wu. et. Al., 2012. A Physical-layer Network Coding Relay scheme for IEEE 802.11.
- [11]. Maxim Integrated Product Inc. MAX2828/2829 Single-/DualBand 802.11a/b/g World-band Transceiver ICs Datasheet. 2004
- [12]. Sklar, Berhard, “ Digital Communications Fundamentals and Applications”. Prentice-Hall.2011
- [13]. miqbal.staff.telkomuniversity.ac.id/?p=344 (Online), (diakses 8 Juni 2015, Jam 13.25)
- [14]. Dereje H. Woldegebreal, Holger Karl. “Network-Coding-based Adaptive Decode and Forward Cooperative Transmission in a Wireless Network: Outage Analysis”. Paderborn. IEEE, April 20



RIWAYAT HIDUP



Penulis bernama lengkap Ummul Khair, anak kelima dari lima bersaudara yang lahir di Padang Ganting pada tanggal 7 Maret 1993. Penulis memulai pendidikan tingkat SD pada tahun 1998 di SDN 08 Padang Ganting, Sumatera Barat. Lulus dari SMPN 1 Padang Ganting pada tahun 2007, dan SMAN 3 Batusangkar pada tahun 2010. Gelar Ahli Madya (D3) diperoleh pada tahun 2013 dari Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Padang. Pada tahun yang sama penulis melanjutkan tingkat Sarjana di Bidang Studi Teknik Telekomunikasi dan Multimedia, Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya. Penulis dapat dihubungi melalui alamat email: ummkha7@gmail.com.