



TUGAS AKHIR - TE 141599

**IMPLEMENTASI DAN ANALISIS KINERJA
NETWORK CODING PADA SISTEM KOMUNIKASI
*ORTHOGONAL FREQUENCY DIVISION
MULTIPLEXING (OFDM)* MENGGUNAKAN
*WIRELESS OPEN-ACCESS RESEARCH (WARP)***

**Bambang Eko Surya
NRP. 2213106008**

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

**JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT - TE 141599

***IMPLEMENTATION AND PERFORMANCE
ANALYSIS OF NETWORK CODING AT
ORTHOGONAL FREQUENCY DIVISION
MULTIPLEXING (OFDM) COMMUNICATION
SYSTEM USING WIRELESS OPEN-ACCESS
RESEARCH PLATFORM MODULE(WARP)***

Bambang Eko Surya
NRP. 2213106008

Supervisors
Dr. Ir. Suwadi, M.T.
Dr. Ir. Wirawan, DEA

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

**IMPLEMENTASI DAN ANALISIS KINERJA
NETWORK CODING PADA SISTEM
KOMUNIKASI ORTHOGONAL FREQUENCY
DIVISION MULTIPLEXING (OFDM)
MENGUNAKAN WIRELESS OPEN-ACCESS
RESEARCH (WARP)**

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

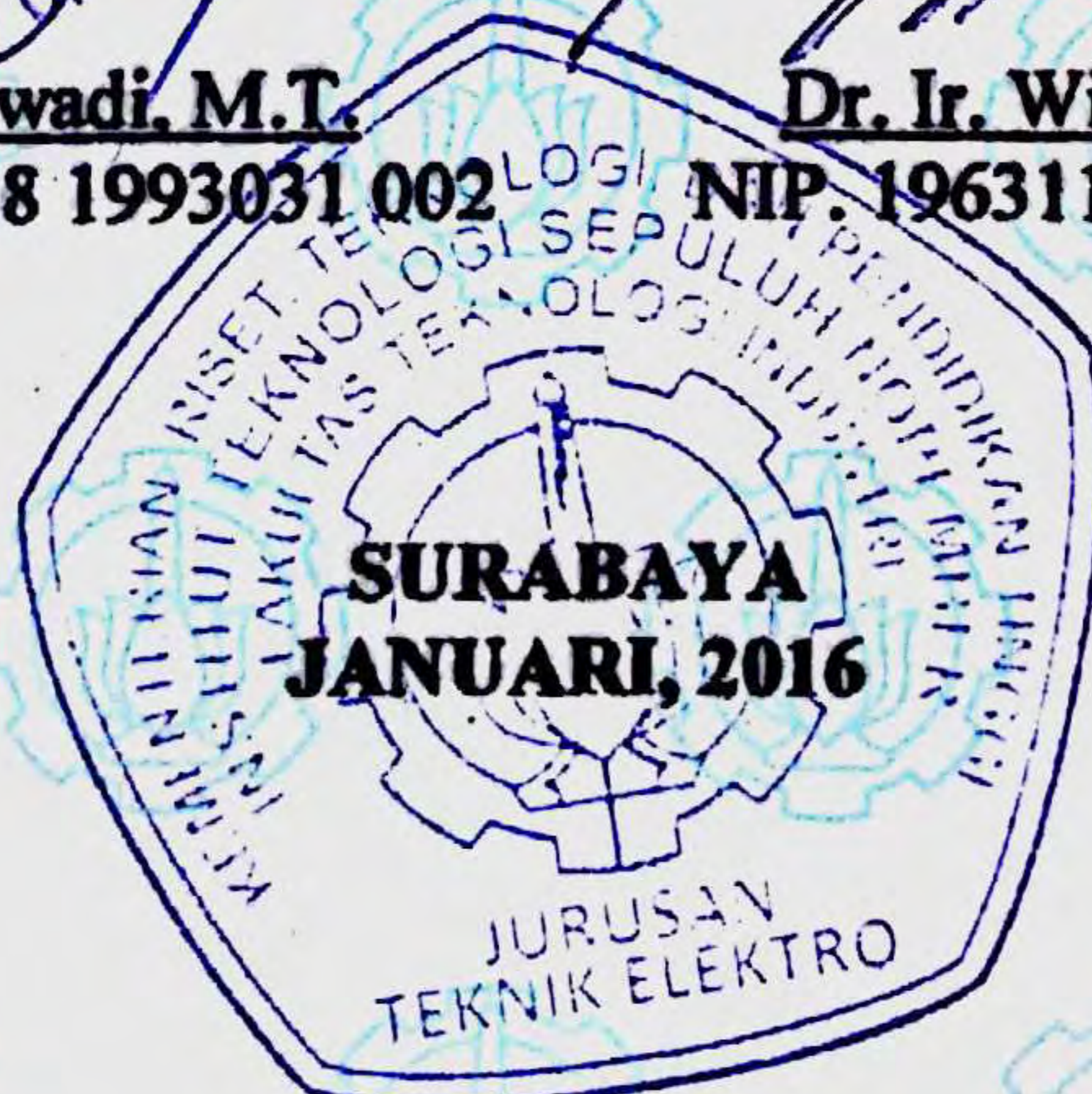
Dosen Pembimbing II

Dr. Ir. Suwadi, M.T.

Dr. Ir. Wirawan, DEA

NIP. 19680818 1993031 002

NIP. 19631109 1989031 011



IMPLEMENTASI DAN ANALISIS KINERJA *NETWORK CODING* PADA SISTEM KOMUNIKASI *ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM)* MENGGUNAKAN *WIRELESS OPEN-ACCESS RESEARCH (WARP)*

Nama : Bambang Eko Surya
Pembimbing : Dr. Ir. Suwadi, M.T.
Dr. Ir. Wirawan, DEA.

ABSTRAK

Network coding di dalam sistem komunikasi semakin populer dan berkembang dikarenakan kemampuannya dapat mengurangi *delay* dan meningkatkan *throughput*. Disaat yang bersamaan, teknik modulasi *Orthogonal Frequency Division Multiplexing (OFDM)* juga semakin populer dan berkembang. Hadirnya OFDM memiliki keuntungan dalam mengefisienkan penggunaan spektrum *bandwidth* dan ketahanan terhadap ISI.

Penggabungan kedua teknologi diharapkan mampu menggabungkan keuntungan – keuntungan dari kedua sistem. Pada tugas akhir, penulis membangun sistem komunikasi OFDM dengan skema *network coding*. Guna memperlihatkan unjuk kerjanya, penulis juga membangun sistem komunikasi OFDM dengan skema *direct* dan *multihop*. Selanjutnya parameter yang dianalisa adalah *bit error rate*, waktu satu siklus transmisi dan uji coba pengiriman citra gambar. Kemudian parameter tersebut dianalisa dan dibandingkan kinerjanya.

Hasil dari pengujian menunjukkan skema *network coding* dan *multihop* konvensional memperbaiki performa BER rata – rata dari skema *direct* sebesar 10dB pada nilai 10^{-3} . improvisasi dari waktu satu siklus transmisi rata – rata 27% lebih cepat , perbaikan *throughput* sebesar $\pm 28\%$ dan perbaikan PSNR pada pengiriman citra yang dikirim.

Kata Kunci : *network coding, OFDM, multihop, direct*

IMPLEMENTATION AND PERFORMANCE ANALYSIS OF NETWORK CODING AT ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM) COMMUNICATION SYSTEM USING WIRELESS OPEN-ACCESS RESEARCH (WARP)

Name : Bambang Eko Surya
Supervisor : Dr. Ir. Suwadi, M.T.
Dr. Ir. Wirawan, DEA.

ABSTRACT

Network coding in communication systems growing more and more popular due to its ability to reduce delay and improve throughput. At the same time, the modulation technique of Orthogonal Frequency Division Multiplexing (OFDM) is also getting popular and growing. The presence of OFDM has an advantage in the use of spectrum bandwidth efficiently and resistance to ISI.

Merging the two technologies is expected to combine the advantages of both systems. In this final project, the authors build OFDM communication systems with network coding scheme. To demonstrate its performance, the authors also construct OFDM communication system with direct and multihop scheme. Parameter measured were the bit error rate, one cycle of transmission and testing of picture image delivery. Then these parameters are analyzed and compared its performance.

Results of the testing showed network coding scheme and the conventional multihop improve BER performance - average of the direct scheme by 10dB on the value of 10^{-3} . Improves of average transmission time of one cycle - average 27% faster, Improves 28% of throughput and PSNR improvement in average delivery posted image average by 11 dB.

Keyword : *network coding, OFDM, multihop, direct*

KATA PENGANTAR

Alhamdu lillaahi rabbil 'aalamiin. Dengan mengucapkan puji syukur kepada Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul :

“IMPLEMENTASI DAN ANALISIS KINERJA *NETWORK CODING* PADA SISTEM KOMUNIKASI *ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM)* MENGGUNAKAN *WIRELESS OPEN-ACCESS RESEARCH (WARP)*”

Penulis tak lupa mengucapkan terima kasih kepada seluruh pihak yang membantu hingga penyusunan tugas akhir ini dapat selesai. Semoga apa yang penulis lakukan dapat memberikan manfaat khususnya bagi pengembangan dan penelitian di bidang telekomunikasi.

Pada kesempatan kali ini tanpa mengurangi rasa hormat, saya ucapkan terima kasih banyak kepada pihak – pihak yang membantu penyelesaian tugas akhir ini, antara lain :

1. Kedua orang tua penulis, Ayahanda Irwan Tulus dan Ibunda Ismiaty, kedua adik – adikku Putri Dwita Mandasari dan Muhammad Ferry Aulia.
2. Pembimbing tugas akhir penulis Bapak Dr. Ir. Suwadi, MT. dan Bapak Dr. Ir. Wirawan, DEA.
3. Ketua Jurusan Teknik Elektro ITS dan Seluruh Bapak/ Ibu dosen
4. Rekan - rekan penelitian *network coding* Bapak Muhammad Iqbal., Prastiyanto, Diefa Agung Riyadi dan Ummul Khair
5. Teman – teman angkatan LJ TMM angkatan gasal 2013
6. Saudara – saudaraku satu kontrakan, Bang Fadli, Kimi, Gaspar, Pepep, Danar, Tiyan dan Depa.

Dan semua pihak yang berjasa dalam Tugas Akhir ini yang penulis tidak dapat sebutkan satu persatu penulis mengucapkan terima kasih sebesar - besarnya.

Surabaya, 16 Desember 2015

Bambang Eko Surya

DAFTAR ISI

ABSTRAK	I
ABSTRACT	III
KATA PENGANTAR	V
DAFTAR ISI	VII
DAFTAR GAMBAR	XI
DAFTAR TABEL	XV
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah.....	2
1.4 Tujuan Tugas Akhir	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penelitian	4
BAB 2 TEORI PENUNJANG	5
2.1 Teknik Orthogonal Frequency Division Multiplexing	5
2.1.1 Mapper dan demapper QPSK.....	7
2.1.2 Pilot.....	8
2.1.2.1 <i>Tipe block</i>	8
2.1.2.2 <i>Tipe comb</i>	9
2.1.2.3 <i>Tipe lattice</i>	10
2.1.3 Inverse Fast Fourier Transform (IFFT) dan Fast Fourier Transform (FFT).....	10
2.1.4 Penambahan dan pembuangan <i>Cyclic Prefix</i> (CP).....	11
2.1.5 Kelebihan dan Kekurangan OFDM.....	12
2.2 Network Coding.....	13
2.2.1 Skema Tradisional Tanpa Network Coding (TS).....	13
2.2.2 Skema Straightforward Network Coding (SNC).....	14
2.3 <i>Wireless Open-Access Research Platform</i>	14
2.3.1 WARP Board	16

2.3.2	WARP Lab	18
2.4	<i>Fresnel Zone</i>	19

BAB 3 PERANCANGAN DAN IMPLEMENTASI. 21

3.1	Alur Perancangan Implementasi	21
3.2	Perancangan sistem pengirim OFDM	22
3.2.1	Pembangkitan informasi	23
3.2.1.1	<i>Pembangkitan informasi secara acak</i>	23
3.2.1.2	<i>Informasi citra gambar</i>	24
3.2.2	Mapping QPSK dan serial to parallel	24
3.2.3	Interpolarisasi	28
3.3	Perancangan Sistem Penerima OFDM.....	30
3.3.1	Desimasi	30
3.3.2	Korelasi LTS.....	31
3.3.3	Estimasi kanal	32
3.3.4	Koreksi kesalahan fase	33
3.3.5	Ekstraksi payload dan <i>serial to parallel</i>	34
3.3.6	Demapping QPSK dan paralel to serial	35
3.4	Perancangan Sistem Komunikasi OFDM <i>Direct</i>	35
3.5	Perancangan sistem komunikasi OFDM <i>multihop</i> konvensional.....	38
3.6	Perancangan sistem komunikasi OFDM denagn <i>network coding</i>	41
3.7	Integrasi PC/Laptop ke perangkat WARP	44
3.8	Skenario Pengukuran	46
3.8.1	Pengukuran lingkungan dalam ruangan.....	48
3.9	Validasi Menggunakan Simulasi Pada Kanal Ideal (<i>Noiseless</i>).....	52
3.9.1	Skema <i>direct</i>	52
3.9.2	Skema <i>multihop</i> konvensional	53
3.9.3	skema <i>network coding</i>	55

BAB 4 HASIL PENGUKURAN DAN ANALISA.... 58

4.1	Hasil dan Analisa Pengukuran BER	58
4.1.1	Pengukuran BER pada kondisi tanpa penghalang.....	58
4.1.2	Pengukuran BER pada kondisi dengan penghalang.....	59

4.1.3	Pengukuran BER pada lingkungan <i>outdoor</i> dan <i>indoor</i>	60
4.1.4	Pengukuran BER skema <i>network coding</i> pada berbagai jarak	61
4.2	Hasil dan Analisa Pengukuran Waktu Komunikasi	62
4.3	Hasil dan Analisa Pengukuran <i>Throughput</i>	63
4.4	Hasil dan Analisa Pengiriman Gambar	64
4.4.1	Pengukuran di dalam ruangan pada kondisi tanpa penghalang.....	64
4.4.2	Pengukuran di dalam ruangan pada kondisi dengan penghalang.....	65
4.4.3	Pengukuran di luar ruangan pada kondisi tanpa penghalang.....	66
4.4.4	Pengukuran di luar ruangan pada kondisi dengan penghalang.....	67
BAB 5	PENUTUP.....	68
5.1	Kesimpulan	68
5.2	Saran	69
	DAFTAR PUSTAKA	70
	LAMPIRAN A : USULAN TUGAS AKHIR	72
	LAMPIRAN B : PROGRAM MATLAB.....	74
	DAFTAR RIWAYAT HIDUP.....	96

DAFTAR GAMBAR

Gambar 2-1. Spektrum frekuensi OFDM	6
Gambar 2-2. Blok diagram OFDM.....	6
Gambar 2-3. Konstelasi QPSK.....	8
Gambar 2-4. Susunan <i>pilot</i> tipe <i>block</i> [4].....	9
Gambar 2-5. Susunan <i>pilot</i> tipe <i>comb</i> [4].....	9
Gambar 2-6. Susunan <i>Pilot</i> tipe <i>Lattice</i> [4].....	10
Gambar 2-7. <i>Cyclic Prefix</i> pada OFDM [4]	12
Gambar 2-8. Komponen-komponen dari Desain Platform [13]	15
Gambar 2-9. Diagram Blok <i>Node</i> WARP	17
Gambar 2-10. Ilustrasi <i>Fresnel zone</i>	20
Gambar 3-1. Alur perancangan sistem	22
Gambar 3-2. Stem simbol informasi.....	23
Gambar 3-3. Konstelasi <i>mapping</i> simbol QPSK $\theta = \pi/2$	25
Gambar 3-4. Ilustrasi konversi <i>serial to parallel</i>	26
Gambar 3-5. Ilustrasi penyalinan <i>cyclic prefix</i>	27
Gambar 3-6 Susunan <i>Preamble</i>	27
Gambar 3-7. Blok diagram Interpolarisasi	28
Gambar 3-8. Respon impuls LPF interpolasi.....	29
Gambar 3-9. Gambaran Output dari setiap blok interpolasi	29
Gambar 3-10. Struktur frame sistem OFDM.....	30
Gambar 3-11. Blok diagram desimasi	30
Gambar 3-12. Gambaran Output dari setiap blok desimasi	31
Gambar 3-13. Konstelasi pada proses estimasi kanal.....	33
Gambar 3-14. Perbandingan Fasa pada Sinyal dengan Koreksi Fasa	34
Gambar 3-16. Proses pengirima dan penerima skema <i>direct</i> pada timeslot I.....	36
Gambar 3-17. Proses pengirima dan penerima skema <i>direct</i> pada timeslot II	37
Gambar 3-18. Skema komunikasi OFDM <i>multihop</i> konvensional	38

Gambar 3-19. Proses pengiriman dan penerima skema <i>multihop</i> pada timeslot I dan II	39
Gambar 3-20. Proses pengiriman dan penerima skema <i>multihop</i> pada timeslot III dan IV	40
Gambar 3-21. Skema komunikasi OFDM <i>network coding</i>	41
Gambar 3-23. Proses pengiriman dan penerima skema <i>network coding</i> pada timeslot III dan IV	43
Gambar 3-24. Integrasi PC/Laptop dengan Modul WARP	44
Gambar 3-25. Pengaturan IP <i>Address</i> pada PC/Laptop.....	44
Gambar 3-26. Konfigurasi <i>dipswitch</i> Untuk Mengatur IP pada WARP.....	45
Gambar 3-27. Pengaturan Tabel Routing.....	45
Gambar 3-28. <i>Set Path</i> Folder WARPLab pada MATLAB.....	46
Gambar 3-29. Konfigurasi <i>Nodes</i> pada MATLAB	46
Gambar 3-30. Grafik radius fresnel pertama fungsi dari jarak antena	47
Gambar 3-31. Dimensi dari penghalang	48
Gambar 3-32. Pengaturan posisi pengukuran tanpa penghalang	49
Gambar 3-33. Pengaturan posisi pengukuran dengan penghalang	49
Gambar 3-34. Gambaran pengukuran lingkungan di dalam ruangan.....	50
Gambar 3-35. Pengaturan posisi pengukuran tanpa penghalang	50
Gambar 3-36. Pengaturan posisi pengukuran dengan penghalang	51
Gambar 3-37. Gambaran pengukuran lingkungan di luar ruangan	51
Gambar 3-38. Konstelasi simulasi skema <i>direct</i> pada kondisi kanal ideal (<i>noiseless</i>)	52
Gambar 3-39 Cuplikan bit informasi simulasi skema <i>direct</i> pada kondisi kanal ideal (<i>noiseless</i>)	53
Gambar 3-40. Konstelasi simulasi skema <i>multihop</i> pada kondisi kanal ideal (<i>noiseless</i>) I.....	53

Gambar 3-41. Konstelasi simulasi skema <i>multihop</i> pada kondisi kanal ideal (<i>noiseless</i>) II.....	54
Gambar 3-42. Cuplikan bit informasi simulasi skema <i>multihop</i> pada kondisi kanal ideal (<i>noiseless</i>)	54
Gambar 3-43. Konstelasi simulasi skema <i>network coding</i> pada kondisi kanal ideal (<i>noiseless</i>) I	55
Gambar 3-44. Konstelasi simulasi skema <i>network coding</i> pada kondisi kanal ideal (<i>noiseless</i>) II	55
Gambar 3-45. Cuplikan bit informasi simulasi skema <i>network coding</i> pada kondisi kanal ideal (<i>noiseless</i>).....	56
Gambar 4-1. Kinerja BER pada kondisi tanpa penghalang	58
Gambar 4-2. Kinerja BER pada kondisi dengan penghalang	59
Gambar 4-3. Kinerja BER pada lingkungan <i>indoor</i> dan <i>outdoor</i>	60
Gambar 4-4. Kinerja BER pada berbagai Jarak.....	61
Gambar 4-5. Waktu satu kali transmisi	62
Gambar 4-6. <i>Throughput</i> dari skema <i>network coding</i> dan <i>multihop</i>	63
Gambar 4-7 . Hasil pengiriman gambar di dalam ruangan dengan kondisi tanpa penghalang	64
Gambar 4-8. Hasil pengiriman gambar di dalam ruangan dengan kondisi dengan penghalang	65
Gambar 4-9. Hasil pengiriman gambar di luar ruangan dengan kondisi tanpa penghalang	66
Gambar 4-10. Hasil pengiriman gambar di luar ruangan dengan kondisi dengan penghalang	67

DAFTAR TABEL

Tabel 3-1 Parameter Desain sistem OFDM.....	21
Tabel 3-2. Nilai desimal dan biner informasi	23
Tabel 3-3. Nilai simbol <i>mapping</i> dari informasi	25
Tabel 3-4. Nilai simbol <i>demapping</i>	35

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada penelitian yang telah ada, *orthogonal frequency division multiplexing* (OFDM) menarik perhatian sebagai teknik transmisi pada jaringan nirkabel modern yang dapat membawa data yang relatif besar. Ada banyak keuntungan dari OFDM, seperti penggunaan yang efektif dari pita frekuensi yang besar, dan ketahanan terhadap *frequency selective fading* [1]. Oleh karena itu OFDM mendapatkan peran pada pengembangan sistem komunikasi nirkabel *broadband* seperti Wi-Fi, WiMax, dan *long term evolution* (LTE) [2].

Pada sistem komunikasi *multihop relay network*, apabila terdapat 2 titik sumber komunikasi (*node* S1 dan S2) dan terdapat 1 buah *node* sebagai perantara (*node* R). jika teknologi transmisinya menggunakan prinsip *time division duplexing* (TDD), maka untuk dapat melakukan pertukaran informasi pada komunikasi *multihop* setidaknya membutuhkan 4 *timeslot*. Diantaranya : *Timeslot* 1, S1 mengirimkan ke R. *Timeslot* 2, R mem-*forward* informasi S1 dengan cara mem-*broadcast* sehingga S2 menerima informasi S1. *Timeslot* 3, S2 mengirimkan ke R. *Timeslot* 4, R mem-*forward* informasi S2 dengan cara mem-*broadcast* sehingga S1 menerima informasi S2. Total membutuhkan 4 *timeslot* agar keduanya dapat berkomunikasi. Dengan menggunakan *network coding*, dibutuhkan hanya 3 *timeslot* [3]. Sehingga dengan *network coding* diharapkan dapat meningkatkan *throughput*.

Network coding melakukan operasi *xor*, pengiriman data cukup dengan melakukan *xor* antara *node* 1 dengan *node* 2 (*coding*), kemudian lakukan *xor* kembali untuk men-*decode* data. Sehingga nantinya dengan *network coding* dapat meningkatkan *throughput* dan mengurangi *delay*.

1.2 Perumusan Masalah

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

1. Bagaimana prinsip kerja *network coding*
2. Bagaimana mengimplementasikan *network coding* pada sistem komunikasi OFDM

3. Seberapa baik kinerja sistem komunikasi OFDM jika ditambahkan network coding

1.3 Batasan Masalah

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

1. Tools simulasi yang digunakan menggunakan perangkat lunak Matlab.
2. Modulasi yang dipakai pada sistem komunikasi OFDM adalah QPSK.
3. Pengimplementasian pada WARP dalam mode non-realtime dengan bantuan WARPLAB 7
4. Parameter yang diukur adalah bit error rate (BER) dan durasi waktu satu cycle .
5. Daerah pengukuran dibatasi dalam lingkungan indoor dan outdoor dengan berbagai macam kondisi perubahan jarak.

1.4 Tujuan Tugas Akhir

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

1. Mengimplementasikan sistem OFDM dengan network coding.
2. Menganalisa kualitas kinerja sistem komunikasi OFDM dengan network coding.
3. Dapat mengimplementasikan sistem OFDM dengan network coding.
4. Mengetahui perbandingan kinerja antara komunikasi OFDM yang menggunakan network coding dengan yang tidak menggunakan network coding.

1.5 Metodologi Penelitian

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

1. Studi Literatur
Langkah pertama yang dilakukan adalah mengumpulkan dan mempelajari literatur yang berhubungan dengan topik tugas akhir ini, yaitu :
 - a. Modulasi digital
 - b. Sistem komunikasi OFDM
 - c. Network coding

- d. Simulasi sistem komunikasi
 - e. Referensi pemrograman modul WARP
 - f. Referensi WARPLab
2. Perancangan Sistem
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*.
 3. Simulasi sistem
Proses simulasi yaitu dengan menggunakan software Matlab. Simulasi bertujuan untuk melihat apakah algoritma yang digunakan sesuai dengan teori sistem komunikasi.
 4. Implementasi pada modul WARP
Sistem komunikasi OFDM menggunakan *network coding* ini akan diimplementasikan pada modul WARP dengan menggunakan perangkat lunak WARPLab
 5. Pengukuran data
Setelah modul WARP bisa digunakan untuk simulasi dengan kondisi kanal yang sebenarnya, 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*.
 6. Proses analisa dan pengambilan kesimpulan
Analisa data dan pengambilan kesimpulan dilakukan setelah pengukuran data selesai. Hal yang akan dianalisa berupa perbandingan kinerja system komunikasi OFDM dengan *network coding* dan tanpa *network coding*.
 7. 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 pada

1.6 Sistematika Penelitian

Proses penelitian yang tersusun dalam laporan tugas akhir ini dapat dijabarkan sebagai berikut :

1. BAB I

Bab 1 berisi latar belakang tugas akhir, tujuan tugas akhir, perumusan masalah, batasan masalah, metodologi dan sistematika penelitian

2. BAB II

Pada bab ini akan dijabarkan mengenai dasar teori yang menjadi acuan dalam pengerjaan pekerjaan ilmiah ini. Pembahasan dalam bab 2 memiliki tujuan antara lain untuk memberikan pemahaman yang mendasar mengenai topik *network coding* dan OFDM. Dan terakhir juga diperlukan pemahaman tentang perangkat yang digunakan sebagai media implementasi, yaitu WARP. Selain pemahaman tentang perangkat keras, juga perlu diketahui tentang pemahaman tentang *framework* WARPLab yang digunakan sebagai pen jembatan Matlab dengan perangkat WARP.

3. BAB III

Pembahasan pada bab tiga terkait dengan sistem *network coding* yang diimplementasikan pada komunikasi OFDM. Hal yang dibangun meliputi desain sistem komunikasi SISO, sistem komunikasi OFDM *multihop* dan yang terakhir komunikasi OFDM dengan *network coding*.

4. BAB IV

Pada bab ini hasil dari simulasi dan implementasi sistem pada WARP diperlihatkan dan kemudian dianalisa. Hasil simulasi dan Pengukuran menghasilkan beberapa parameter yang diukur meliputi *bit error rate*, waktu satu *cycle* transmisi dan memperlihatkan gambar hasil transmisi baik itu skema komunikasi *direct*, *multihop* konvensional dan *network coding*.

5. BAB V

Pada bagian ini akan paparkan mengenai kesimpulan dari hasil penelitian dan juga saran – saran yang terkait mengenai penelitian ini

BAB 2 TEORI PENUNJANG

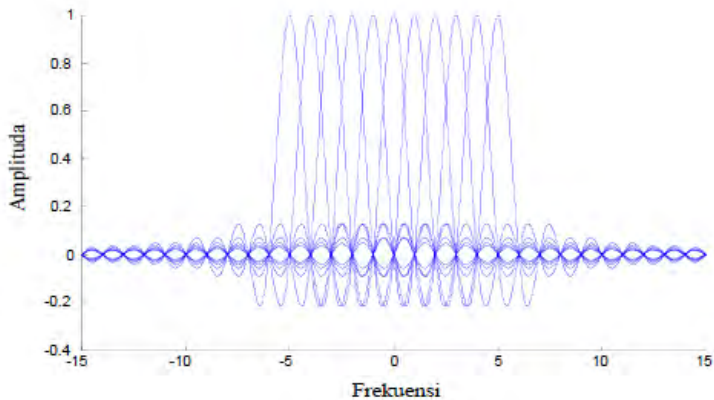
Pada bab ini akan dijabarkan mengenai dasar teori yang menjadi acuan dalam pengerjaan pekerjaan ilmiah ini. Pembahasan dalam bab 2 memiliki tujuan antara lain untuk memberikan pemahaman yang mendasar mengenai topik *network coding* dan OFDM. Dan terakhir juga diperlukan pemahaman tentang perangkat yang digunakan sebagai media implementasi, yaitu WARP. Selain pemahaman tentang perangkat keras, juga perlu diketahui tentang pemahaman tentang *framework* WARPLab yang digunakan sebagai penjembutan Matlab dengan perangkat WARP.

2.1 Teknik Orthogonal Frequency Division Multiplexing

Orthogonal frequency division multiplexing atau selanjutnya disebut OFDM adalah bentuk dari *multi-carrier modulation* (MCM) yang dapat dapat dicapai dengan membagi kanal transmisi tunggal menjadi beberapa *subchannel* atau *subcarrier* yang saling tegak lurus (*orthogonal*) untuk mengoptimalkan efisiensi transmisi data. Sinyal *carrier* dari OFDM merupakan penjumlahan dari banyaknya *subcarrier* yang orthogonal, dengan data *baseband* pada masing – masing *subcarrier* dimodulasikan dengan teknik modulasi yang diinginkan. Pemakaian frekuensi yang saling orthogonal pada sistem OFDM memungkinkan singgungan antar frekuensi tanpa menimbulkan interferensi antara satu dengan lainnya. Dalam pendekatan matematis [4], keorthogonalan sistem OFDM dapat dijelaskan sebagai berikut, misalkan fungsi $X_m(t)$ adalah *subcarrier* ke- m dan $X_n(t)$ adalah *subcarrier* ke- n . Kedua *subcarrier* ini dikatakan saling orthogonal satu sama lain pada interval $a < t < b$ jika memenuhi kondisi :

$$\int_a^b X_m(t) X_n^*(t) dt = 0 \quad (2.1)$$

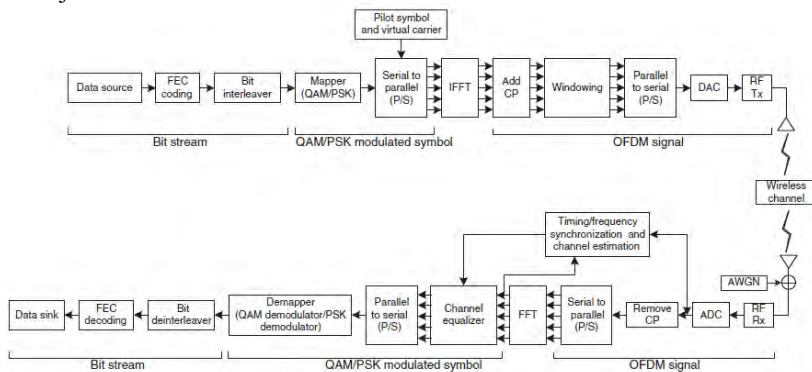
Spektrum frekuensi yang saling orthogonal OFDM diperlihatkan pada **Gambar 2-1**. Terlihat setiap frekuensi *subcarrier* saling bersinggungan.



Gambar 2-1. Spektrum frekuensi OFDM

Kaedah utama dari sistem OFDM adalah dimana skema modulasinya dengan rasio simbol yang rendah sehingga hanya mendapat sedikit dari pengaruh *intersymbol interference* dari *multipath fading*. Oleh karena itu, maka dapat ditransmisikan sejumlah aliran *low-rate* dalam bentuk parallel, bukan aliran *high-rate* seri atau tunggal. Karena durasi simbol lebih panjang jika dibandingkan dengan aliran *high-rate*, maka memungkinkan untuk penyisipan *guard interval* diantara simbol – simbol OFDM, sehingga dapat menghilangkan *intersymbol interference*.

OFDM terbentuk dari beberapa bagian, diantaranya seperti yang ditunjukkan



Gambar 2-2. Blok diagram OFDM

2.1.1 Mapper dan demapper QPSK

Bit-bit informasi akan dibangkitkan secara acak dan berupa data digital (1 dan 0) dimodulasi phase digital, bentuk gelombang sinyal M digambarkan sebagai [5]:

$$\begin{aligned} s_m(t) &= \operatorname{Re} \left[g(t) e^{j\frac{2\pi(m-1)}{M}} e^{j2\pi f_c t} \right], \quad m = 1, 2, \dots, M \quad (2.2) \\ &= g(t) \cos \left[2\pi f_c t + \frac{2\pi}{M}(m-1) \right] \\ &= g(t) \cos \left(\frac{2\pi}{M}(m-1) \right) \cos 2\pi f_c t \\ &\quad - g(t) \sin \left(\frac{2\pi}{M}(m-1) \right) \sin 2\pi f_c t \end{aligned}$$

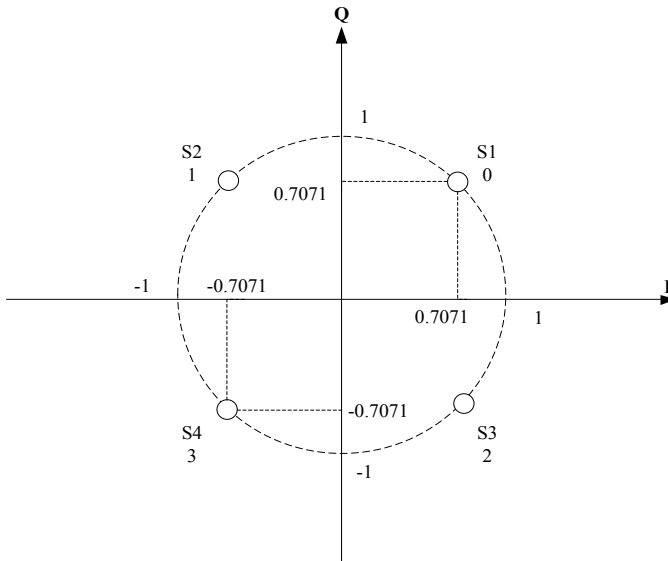
Dimana:

$g(t)$ = Bentuk sinyal pulsa

θ_m = $2\pi(m-1)/M, m = 1, 2, \dots, M$

QPSK merupakan teknik pengkodean M-ary dimana $M = 4$. M berarti digit yang mewakili banyaknya kondisi yang mungkin. Dalam modulasi QPSK ada 4 *phase output* yang berbeda, maka harus ada 4 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 ada 4 kondisi yang mungkin yaitu: 00, 01, 10 dan 11 [6]. Singkatnya dalam modulasi QPSK menghasilkan output berupa symbol dimana dalam 1 symbol mewakili 2 bit. Contoh gambar konstelasi QPSK yang mengalami pergeseran fasa 45° diperlihatkan pada **Gambar 2-3**.

Sementara untuk proses demodulasi berkebalikan dari modulasi, yaitu perubahan dari symbol-symbol menjadi suatu bit. Dimana 1 symbol mewakili 2 bit



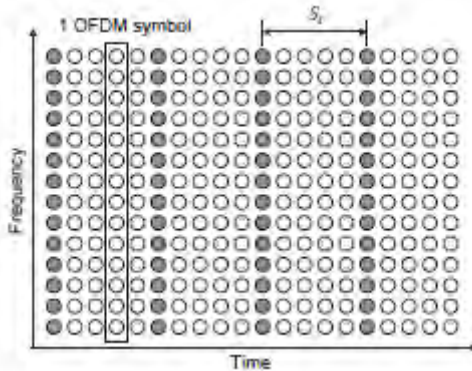
Gambar 2-3. Konstelasi QPSK

2.1.2 Pilot

Pilot merupakan sebuah symbol yang ditransmisikan pada *predefined-sucarrier*. *Pilot* digunakan untuk mengestimasi fasa noise pada penerima [7]. Struktur *Pilot* memiliki 3 tipe, diantaranya yaitu tipe *block*, *comb* dan *lattice* [4].

2.1.2.1 Tipe block

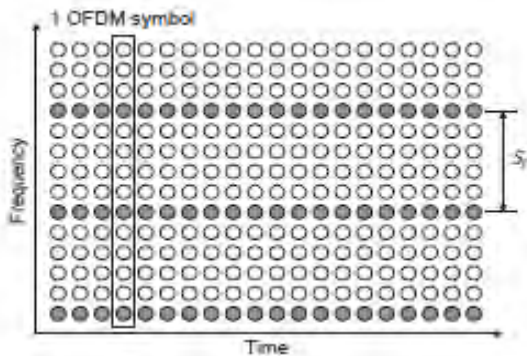
Pada tipe ini, simbol-simbol OFDM dengan *Pilot-Pilot* pada semua *subcarrier* ditransmisi secara periodic untuk kanal estimasi. Susunan *Pilot* tipe ini diperlihatkan pada [8] **Gambar 2-4**. Dimana S_t merupakan periode dari *Pilot* symbol dalam domain waktu. Menggunakan tipe *Pilot* ini, interpolasi dalam domain waktu digunakan untuk mengestimasi kanal sepanjang sumbu x (waktu) [4]. Karena *Pilot* ditempatkan pada semua *subcarrier Pilot symbol* dalam domain waktu, maka tipe *block* ini cocok untuk kanal *frequency selective*.



Gambar 2-4. Susunan *pilot* tipe *block* [4].

2.1.2.2 Tipe *comb*

Pada tipe ini, setiap *symbol* OFDM memiliki *symbol Pilot* pada *subcarrier-subcarrier* yang ditempatkan secara periodic, yang mana digunakan untuk interpolasi dalam domain frekuensi untuk estimasi kanal sepanjang sumbu frekuensi. Susunan *Pilot* tipe ini diperlihatkan pada **Gambar 2.5**.

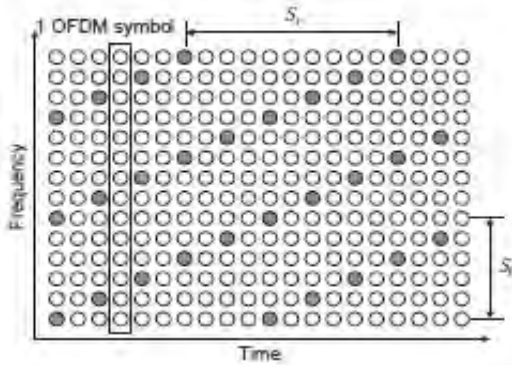


Gambar 2-5. Susunan *pilot* tipe *comb*[4]

Dimana S_f merupakan periode dari *pilot* dalam domain frekuensi. Tipe susunan *Pilot* ini cocok untuk kanal *fast fading* tetapi tidak untuk kanal *frequency selective*.

2.1.2.3 Tipe lattice

Pada tipe ini, Pilot ditaruh baik dalam domain waktu maupun frekuensi dengan periode tertentu. Tujuannya yaitu untuk interpolasi dalam domain waktu/frekuensi untuk estimasi kanal. Susunan Pilot ini diperlihatkan pada Gambar 2.6. Dimana S_t dan S_f menggambarkan periode dari symbol-symbol Pilot dalam domain waktu dan frekuensi.



Gambar 2-6. Susunan *Pilot* tipe Lattice [4]

2.1.3 Inverse Fast Fourier Transform (IFFT) dan Fast Fourier Transform (FFT)

IFFT dan FFT merupakan kunci utama dalam OFDM. Blok IFFT berfungsi untuk membangkitkan frekuensi *carrier* yang saling orthogonal satu sama lain. Jumlah titik IFFT pada implementasi bernilai 2^n , dengan n adalah bilangan bulat positif. Titik IFFT bisa diset sesuai dengan jumlah *subcarrier* yang digunakan atau lebih besar. Dalam implementasi di WARP nanti, jumlah titik IFFT merupakan jumlah dari *subcarrier* yang berisi data input dan *guard subcarrier* yang berisi data dengan nilai 0. Sementara blok FFT berfungsi sebagai osilator local pada penerima yang akan memisahkan antara frekuensi *carrier* dengan symbol-symbol OFDM yang berbeda pada frekuensi tersebut. Jumlah titik FFT sama seperti jumlah titik IFFT. Berdasarkan [5], persamaan untuk proses IFFT pada pemancar untuk *sample* ke- n di symbol OFDM ke- m yaitu:

$$s_{m,n} = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} A_{m,k} e^{j2\pi \frac{kn}{N}}, 0 \leq n \leq N-1 \quad (2.3)$$

Dimana:

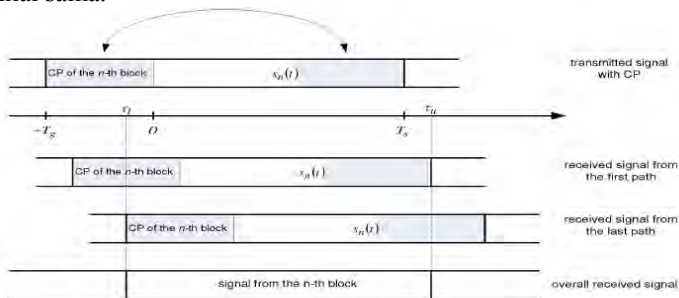
- N = Ukuran IFFT
- $A_{m,k}$ = Symbol data hasil modulasi pada *subcarrier* ke- k di symbol OFDM ke- m

2.1.4 Penambahan dan pembuangan *Cyclic Prefix* (CP)

Cyclic prefix merupakan salinan bagian akhir dari suatu symbol OFDM dan menempatkannya pada bagian depan blok symbol tersebut. Tujuan dari adanya *cyclic prefix* yaitu untuk mengurangi kemungkinan terjadinya ISI bukan berarti menghilangkan. Panjang dari *cyclic prefix* harus lebih panjang daripada kanal respons impuls kanal [9]. Untuk lebih jelasnya fungsi dari *cyclic prefix* diperlihatkan pada Gambar 2.7.

Sementara untuk ilustrasi susunan *cyclic prefix* pada OFDM diperlihatkan pada Gambar 2.8. Dimana l merupakan sampel terakhir dari suatu symbol OFDM, yang kemudian di *copy* dan ditempatkan kebagian awal dari suatu symbol OFDM tersebut. Selain itu, adanya *cyclic prefix* dapat digunakan untuk mendeteksi awal dari masing-masing symbol OFDM. Hal ini dapat dilakukan karena l sampel awal dan terakhir berisi nilai yang sama sehingga dapat dikorelasikan [9].

Data *output* merupakan data yang keluar setelah melalui proses OFDM diatas. Diharapkan nilai data *output* dan nilai data *input* akan bernilai sama.



Gambar 2. 1 Fungsi dari *Cyclic Prefix* [10]

Dari gambar diatas dibuat suatu persamaan tentang panjang keseluruhan dari symbol OFDM yaitu:

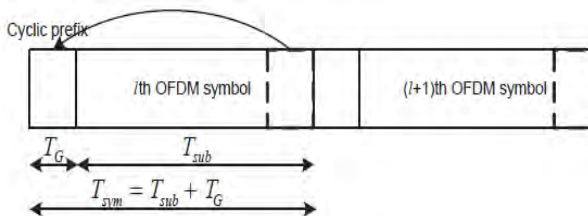
$$T_{sym} = T_{sub} + T_G \quad (2.3)$$

Dimana:

T_{sub} = Panjang symbol OFDM tanpa *cyclic prefix*

T_G = Panjang *cyclic prefix*

T_{sym} = Total panjang symbol OFDM



Gambar 2-7. *Cyclic Prefix* pada OFDM [4]

2.1.5 Kelebihan dan Kekurangan OFDM

Berdasarkan [9], OFDM memiliki beberapa kelebihan dan kerugian jika dibandingkan dengan sistem modulasi *single carrier*.

1. Kelebihan:
 - a. Tahan terhadap penyebaran *delay* akibat multipath; karena durasi symbol dibuat lebih besar.
 - b. Tahan terhadap kanal *frequency selective fading*; karena kanal *bandwidth* yang tersedia dikonversi menjadi beberapa *subcarrier* yang sempit. Sehingga bisa diasumsikan *subcarrier* hanya mengalami *flat fading*
 - c. Efisien dalam hal modulasi dan demodulasi; Modulasi dan demodulasi setiap *subcarrier* hanya menggunakan metode IFFT dan FFT. Dengan menggunakan metode tersebut membuat efisien penggunaan spectrum dengan cara saling *overlap* antar *subcarrier*.
 - d. Transmisi *bit rate* yang tinggi

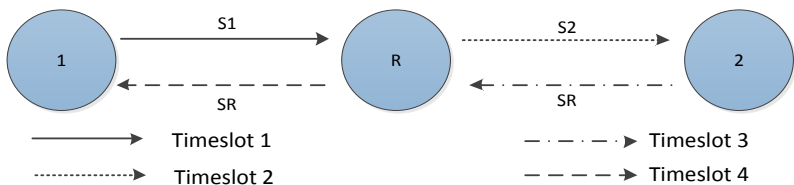
- e. *Equalizer* yang sederhana; Simbol OFDM lebih besar dari maksimum penyebaran *delay* sehingga menghasilkan kanal *flat fading* yang dapat di *equalizer* dengan mudah
 - f. Efisiensi spectral yang besar
2. Kekurangan:
- a. Keakuratan sinkronisasi yang tinggi
 - b. *Peak to Average Power Ratio* (PAPR) yang tinggi
 - c. Lebih sensitive terhadap *carrier offset* dan *drift* dibandingkan dengan sistem *single carrier*.

2.2 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 dikombinasi dengan pengkodean dari beberapa source dengan satu pengiriman [11]. *Network coding* dapat meningkatkan *throughput*, ketahanan, kompleksitas, dan keamanan. Terdapat 3 konsep *network coding* yaitu skema *tradisional* tanpa *network coded* (TS), skema *straightforward network coding* (SNC) dan skema *physical layer network coding* (PNC) [12].

2.2.1 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 2.6 [12].



Gambar 2.2 Tradisional Tanpa Network Coding

2.2.2 Skema Straightforward Network Coding (SNC)

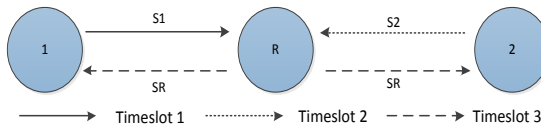
Pada *straightforward network coding*, menggunakan timeslot 3 sehingga dapat memiliki kemampuan untuk menaikkan throughput sebesar 33%. Pada gambar 2.7 menggambarkan ilustrasi SNC menggunakan 3 timeslot untuk proses pengiriman data informasi. Pada timeslot 1, node 1 mengirimkan S_1 ke *relay*. Kemudian pada timeslot 2, node 2 mengirimkan informasi S_2 ke *relay*. Setelah menerima informasi dari S_1 dan S_2 , *relay* kemudian membantuk pemetaan *network coding* $S_R = f(S_1, S_2)$ Sedangkan pada S_1, S_2 terdiri dari simbol modulasi dan $f(S_1, S_2)$ merupakan XOR, sebagai berikut :

$$S_R = S_1 \oplus S_2 \quad (2.4)$$

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 :

$$S_1 \oplus S_2 = S_1 \oplus (S_1 \oplus S_2) = S_2 \quad (2.5)$$

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 [12].



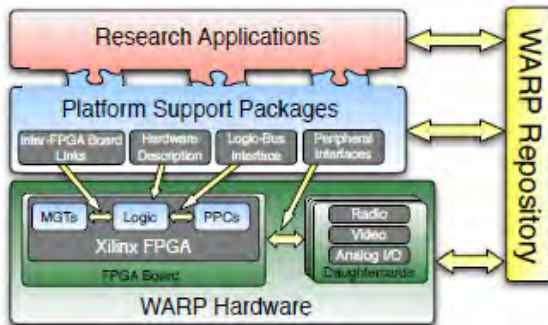
Gambar 2.3 Skema *Straightforward Network Coding*

2.3 *Wireless Open-Access Research Platform*

WARP merupakan suatu *platform wireless* yang dapat diprogram, dapat melakukan *prototype* jaringan *wireless* dari yang dasar hingga kompleks. WARP menggabungkan *hardware* performa tinggi yang dapat diprogram dengan *open-source repository* dari desain-desain referensi dan bahan-bahan sumber [13].

Proyek WARP dimulai pada tahun 2006 oleh Prof. Ashu Sabharwal di Universitas Rice, Amerika [13]. Sebelumnya, proyek ini didanai oleh *National Science Foundation* dengan bantuan langsung dari Xilinx [13].

Proyek WARP telah tumbuh menjadi sebuah usaha mandiri yang *open source* dengan pengguna di seluruh dunia. Dalam penggunaan WARP pada penelitian, terdapat beberapa metode untuk pemrograman perangkat tersebut. Platform ini terdiri dari *hardware* dan implementasi FPGA yang menjadi kunci komunikasi antar blok-blok. *Hardware* pada WARP terdiri dari *FPGA-based processing boards* yang digabungkan untuk radio band lebar (*wideband*) dan *I/O interfaces*. Arsitektur dari WARP diperlihatkan pada **Gambar 2-8** yang terdiri dari 4 komponen utama.



Gambar 2-8. Komponen-komponen dari Desain Platform [13]

1. *Custom Hardware*: dibutuhkan untuk komunikasi *wireless* performa tinggi. *Hardware* menyediakan sumber daya yang *capable* untuk DSP kecepatan tinggi - implementasi algoritma, *scalable* antar koneksi-koneksi untuk mengalokasikan lebih banyak daya pemrosesan yang dibutuhkan dan *extensible* untuk radio, terutama bagian prosesor dan interface ekspansi lainnya.
2. *Platform Support Packages*: mendesain perangkat-perangkat dan modul *interface* level rendah yang didesain untuk memungkinkan penggunaan *hardware* oleh para peneliti pada semua layer desain jaringan *wireless*.
3. *Open-Access Repository*: penyimpanan pusat untuk semua *source code*, model-model dan file-file desain *hardware*. Penyimpanan pusat tersebut dibawah lisensi BSD-like *open-source*.

4. *Research Applications*: pengimplementasian dari algoritma-algoritma, dapat dilakukan dengan cara custom *hardware* dan *platform support packages*. *Full system* dengan algoritma-algoritma tersebut dapat dengan cepat dibangun menggunakan blok modul standar yang disediakan pada *repository*.

Dalam proses desain, setidaknya terdapat dua metode yang bisa digunakan untuk meneliti desain *physical layer*. Metode pertama dengan menggunakan sistem *non-real-time* yaitu dengan WARPLab. Sistem ini menggunakan kondisi kanal *real-time*, namun dengan pemrosesan data secara *offline* melalui MATLAB [8].

Metode kedua adalah dengan membangun sistem *real-time* yang diinginkan untuk diimplementasikan pada FPGA (*Field Programmable Gate Array*). Sistem tersebut bisa dibangun secara independen dan berdiri sendiri pada *node* WARP dan menggunakan *system generator* [8].

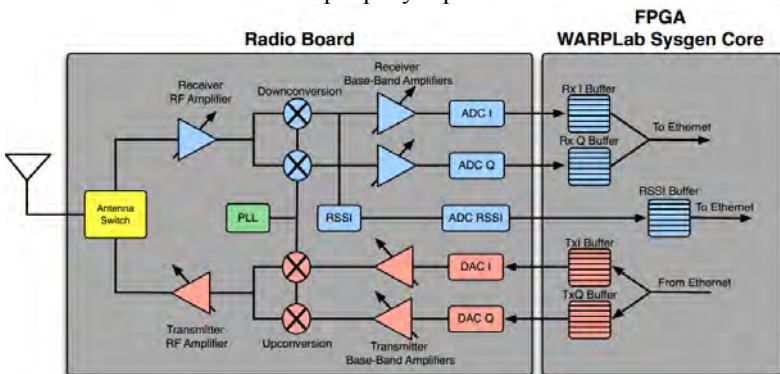
2.3.1 WARP Board

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 dapat dijelaskan dengan blok diagram yang ditunjukkan pada **Gambar 2-9**.

Berdasarkan Gambar 2.14 tersebut dapat terlihat bahwa ada 2 komponen utama pada *node* WARP yaitu radio *board* dan FPGA. Sementara untuk warna merah muda menandakan blok pemancar dan warna biru menandakan blok penerima.

1. Blok pada jalur pemancar:
 - a. Tx I/Q Buffer: tempat penyimpanan sampel *inphase* dan *quadrature* yang akan dikirim.
 - b. DAC I/Q: mengkonversi dari digital ke analog untuk sample *inphase/quadrature*.
 - c. Tx BB Amplifier: kepanjangan dari *Transmitter Base Band Amplifier*. Terdapat dua amplifier didalamnya yaitu satu untuk sinyal *inphase* dan satu untuk sinyal *quadrature*. Ini merupakan *variable gain amplifier* dimana nilainya dapat dimasukkan oleh *user* di *workspace* matlab.
 - d. *Upconversion*: Mengubah sinyal *baseband* ke sinyal RF. Frekuensi *carrier* tergantung pada pengaturan PLL.

- e. Tx RF Amplifier: *Transmitter RF amplifier, Variabel gain amplifier*. Nilai ini yang dimasukkan oleh user di workspace Matlab.
2. Blok pada jalur penerima:
- a. Rx RF Amplifier: Receiver RF amplifier, variable gain amplifier. Nilai ini yang dimasukkan oleh user di workspace Matlab.
 - b. Downconversion: mengubah sinyal RF ke sinyal baseband. Sinyal baseband tersebut terdiri dari sinyal inphase dan quadrature. Downconversion dari RF ini bergantung kepada setting PLL.
 - c. Rx BB Amplifiers: Receiver Base Band Amplifier. Amplifier ini terdiri dari 2 bagian, yaitu satu untuk sinyal inphase dan satu untuk sinyal quadrature. Bagian ini adalah variable gain amplifier, keduanya diset dengan nilai gain sama yang dapat dimasukkan oleh user menggunakan Matlab.
 - d. RSSI: kepanjangan dari Receive Signal Strength Indicator. Blok ini mengukur nilai dari RSSI.
 - e. ADC I/Q: mengkonversi sinyal analog ke digital untuk sample inphase dan quadrature.
 - f. ADC RSSI: mengkonversi sinyal digital ke analog untuk pengukuran RSSI. Data RSSI tersedia di $\frac{1}{4}$ laju data I/Q.
 - g. Rx I/Q Buffer: tempat penyimpanan sample yang diterima baik dalam bentuk inphase maupun quadrature.
 - h. RSSI Buffer: tempat penyimpanan data RSSI.



Gambar 2-9. Diagram Blok *Node WARP*

Pengaturan PLL bertujuan untuk menentukan frekuensi pembawa yang dapat diatur ke salah satu dari 14 kanal wi-fi di band 2.4 GHz atau 23 saluran di 5 GHz. Pengaturan ini dapat diatur dari *workspace* matlab.

2.3.2 WARPLab

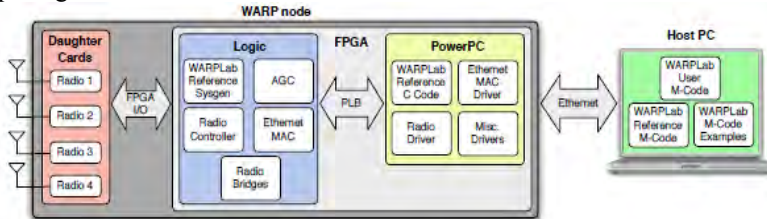
WARPLab merupakan sebuah kerangka kerja yang digunakan untuk *physical layer*. Penggunaan dari WARPLab memungkinkan kombinasi dari beberapa sistem multi-antena pada pemancar dan penerima.

Beberapa hal yang berkaitan dengan WARPLab yaitu [14]:

1. Satu PC/laptop dapat mengontrol banyak node-node WARP.
2. MATLAB untuk pemrosesan sinyal.
3. Pemrosesan tidak *real time*.
4. WARP untuk *interface wireless*.
5. Penggunaan *channel* secara *real time*.

Sample yang ditulis pada MATLAB akan disimpan di FPGA. *Sample* di FPGA akan dikirim melalui udara menggunakan radio-radio yang tersedia.

Secara keseluruhan, arsitektur dari WARPLab diperlihatkan pada gambar 2.15 dibawah ini:

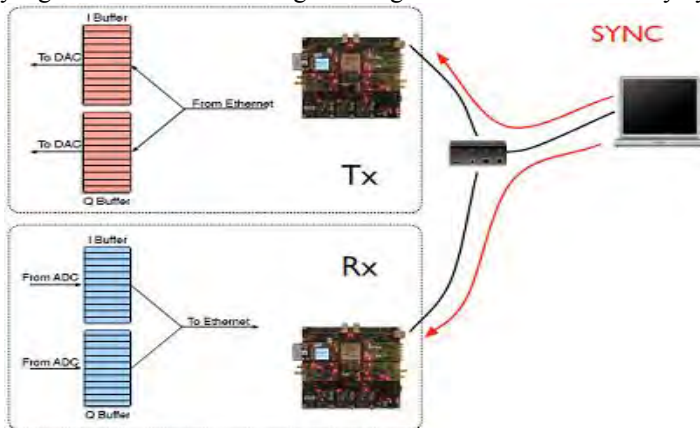


Gambar 2. 4 Arsitektur WARPLab [14]

Sementara untuk aliran kerja dari WARPLab diperlihatkan pada Gambar 2.13 dibawah ini. Seperti yang diperlihatkan pada Gambar 2.13, aliran kerja dari WARPLab yaitu [14]:

1. Inisialisasi node-node dan setting radio.
2. Download vector Tx.
3. Enable jalur Tx/Rx.
4. Mesin utama Tx/Rx dalam keadaan diam.
5. Men-trigger transmisi dan melakukan capture.
6. Menerima vector Rx.

Saat ini, versi WARP dan versi WARPLab yang digunakan pada tugas akhir ini adalah versi 2 dan 7. WARPLab 7 memiliki spesifikasi yang lebih terbaru dibandingkan dengan WARPLab sebelumnya yaitu:



Gambar 2. 5 WARPLab Flow [14]

1. Paket “sniffing” logika (compatible dengan WARP v3)
2. 10 x pengurangan jitter „node-to-node” dalam memulai siklus Tx/Rx
3. Siklus Tx/Rx mulai lebih cepat yaitu sekitar 12 μ s
4. Mendukung untuk frame Ethernet sampai sekitar 9014 byte, sehingga meningkatkan performance secara signifikan.
5. Default buffer sekitar 215 (32 k) sampel (durasi 800 μ s per siklus Tx/Rx) (hanya compatible dengan WARP v3).

2.4 Fresnel Zone

Sinyal nirkabel dapat dipengaruhi oleh beberapa faktor konsisi dan posisi. Salah satu dari faktor tersebut berkaitan dengan kekuatan sinyal adalah ada atau tidaknya gangguan terhadap *fresnel zone*. Gangguan terhadap zona ini bisa berpengaruh menghasilkan fenomena berupa refraksi, difraksi dan refleksi.

Fresnel zone merupakan daerah dimana suatu lintasan transmisi gelombang radio yang digambarkan berbentuk elips yang menunjukkan wilayah interferensi jika terdapat gangguan. Adapun besaran fresnel zone merupakan fungsi dari jarak antara titik pengirim dan penerima dan frekuensi kerja sistem. Rumus perhitungan fresnel tersebut tertera pada **persamaan 2.6**. Ilustrasi dari fresnel zone dapat diperlihatkan pada **Gambar 2-10**.

(3.6)

$$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}$$

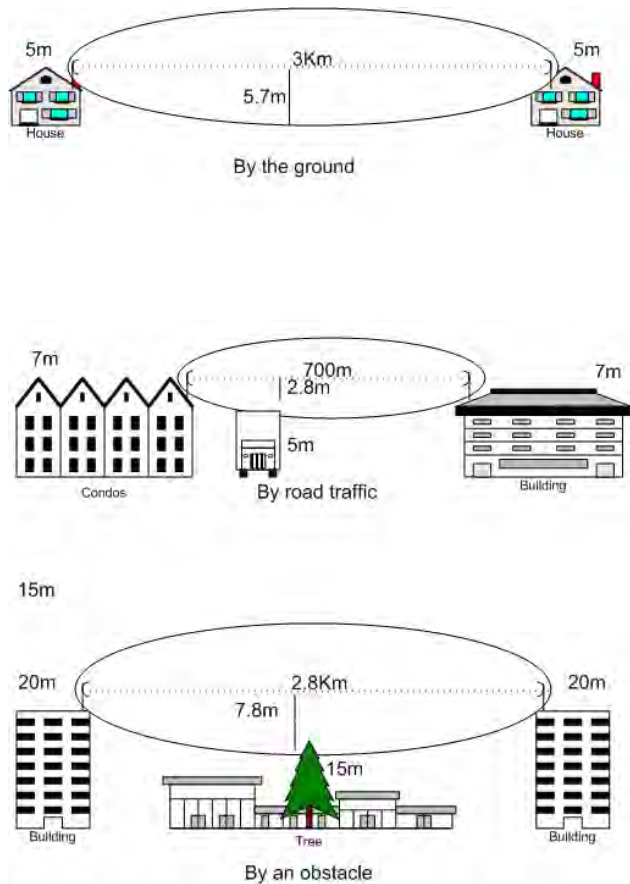
Dimana :

F : Jari jari fresnel

λ : panjang gelombang

d1 : jarak dari pengirim ke titik jari – jari F

d2 : jarak dari penerima ke titik jari – jari F



Gambar 2-10. Ilustrasi *Fresnel zone* [15]

BAB 3

PERANCANGAN DAN IMPLEMENTASI

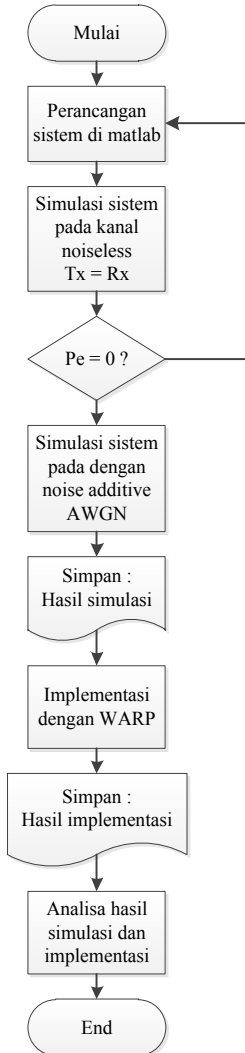
Pembahasan pada bab tiga terkait dengan sistem *network coding* yang diimplementasikan pada komunikasi OFDM. Hal yang dibangun meliputi desain sistem komunikasi SISO OFDM – sistem komunikasi OFDM langsung antar *node* pengirim dan penerima tanpa *relay*, sistem komunikasi OFDM *multihop* konvensional – sistem komunikasi dua arah dengan 4 *slot* waktu pengiriman/penerimaan antara sumber 1 dan sumber 2 dengan melewati *relay* dan yang terakhir komunikasi OFDM dengan *network coding* – sistem komunikasi dua arah dengan 3 *slot* waktu pengiriman/penerimaan antara sumber 1 dan 2 yang melewati *relay* dengan sistem *network coding*.

Tabel 3-1 Parameter Desain sistem OFDM

Parameter		Besaran
Besar FFT		64
Besar CP		16
Jumlah <i>Subcarrier</i>	<i>Payload</i>	48
	<i>Pilot</i>	4
	<i>Null</i>	12
	total	64
interpolarisasi		2 kali
<i>Mapping Baseband</i>		QPSK

3.1 Alur Perancangan Implementasi

Dalam melakukan perancangan implementasi perlu dilakukan beberapa tahapan. Langkah pertama desain sistem, simulasi pada matlab. Simulasi pada matlab dilakukan pengaplikasian pada kondisi *noiseless* dengan tujuan untuk memastikan sistem berjalan benar. Indikasi dari benarnya tidak adalah dengan memperhatikan kondisi informasi yang dikirim dan diterima sama keseluruhan. Lalu disimulasikan menggunakan noise *additive* AWGN. Pada tahap ini ingin memperlihatkan gambaran awal respon BER terhadap noise tambahan AWGN untuk nantinya akan dibandingkan dengan hasil implementasi. Langkah ketiga yaitu pengimplementasian sistem pada WARP, selanjutnya akan dipeoleh nilai BER untuk dia Analisis kinerjanya. Adapun alur perancangan sistem ditunjukkan pada **Gambar 3-1**.



Gambar 3-1. Alur perancangan sistem

3.2 Perancangan sistem pengirim OFDM

Pada bagian pengirim OFDM ini terdiri dari beberapa bagian seperti yang sudah dijelaskan sebelumnya pada bab 2 mengenai bagian –

bagian yang menyusun sistem OFDM. Diantaranya adalah pembangkit informasi, *mapping* QPSK, *serial to parallel*, penyisipan *pilot*, IFFT, *paralel to serial*, penambahan *preamble* dan CP kemudian Interpolarisasi.

3.2.1 Pembangkitan informasi

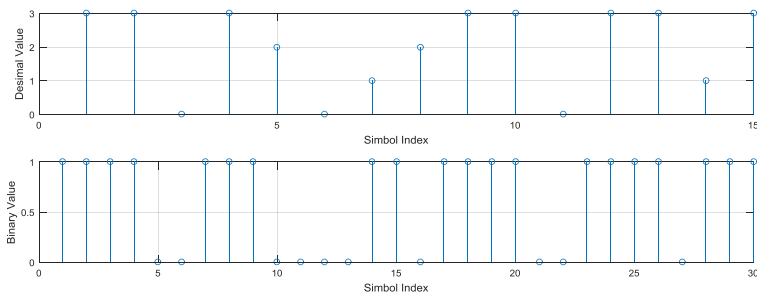
Informasi yang akan dikirimkan pada tugas akhir ini berupa bit yang dibangkitkan secara acak dan informasi berupa citra gambar *grayscale*. Untuk itu dua informasi ini masing masing akan dibangkitkan dengan sistem tersendiri.

3.2.1.1 Pembangkitan informasi secara acak

Data acak akan dibangkitkan sejumlah tertentu. Pada tugas akhir ini data acak yang dibangkitkan bernilai desimal 0 sampai 3 (sesuai dengan jumlah nilai simbol dalam *mapping* QPSK. satu data yang dibangkitkan dari proses acak ini merupakan satu simbol *mapping* yang artinya 1 simbol terdiri dari 2 bit informasi seperti yang diperlihatkan **Tabel 3-2**. Yang mana bit informasi tersebut adalah nilai biner dari data desimal tersebut. Sehingga hasil dari bagian ini adalah informasi yang akan dikirimkan seperti yang ditunjukkan pada **Gambar 3-2**.

Tabel 3-2. Nilai desimal dan biner informasi

Nilai desimal	Nilai biner
0	0 0
1	0 1
2	1 0
3	1 1



Gambar 3-2. Stem simbol informasi

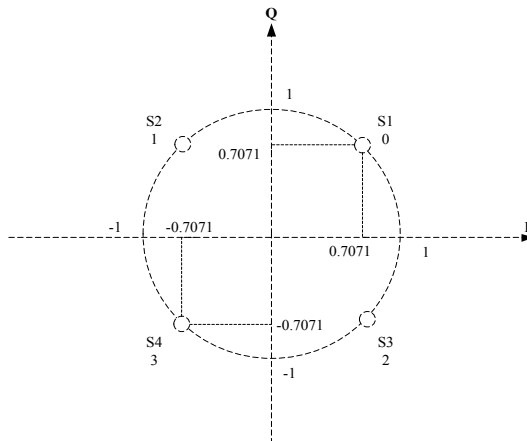
3.2.1.2 Informasi citra gambar

Selain informasi yang akan dikirimkan berupa informasi yang dibangkitkan secara acak, tetapi sistem juga dapat mengirimkan informasi berupa citra gambar. Pada perancangan program bagian ini akan *generate* nilai biner dari citra gambar untuk nantinya akan diolah sebagai informasi yang akan dikirim. Pada tugas akhir ini citra yang dikirim berupa citra *grayscale*. Nilai dari setiap *pixel* diterjemahkan menjadi 8 bit informasi. Kemudian nilai biner dari keseluruhan *pixel* tersebut akan disusun serial dan setiap dua bit informasi akan diubah menjadi desimal agar menjadi simbol informasi yang akan digunakan pada tahap selanjutnya yaitu proses *mapping* QPSK.

3.2.2 Mapping QPSK dan serial to parallel

Dari seriap simbol – simbol informasi akan *dimapping* kedalam konstelasi QPSK $\theta = \pi/2$ dengan masing – masing informasi akan diubah nilainya menjadi bilangan kompleks. Pada

Gambar 3-3 diperlihatkan nilai dari setiap simbol akan *dimapping* kedalam konstelasi QPSK $\theta = \pi/2$ dengan masing – masing nilainya memiliki nilai pada sumbu *in-phase* dan *quadratur*. **Tabel 3-3** menunjukkan besaran nilai – nilai dari setiap *mapping* dari setiap 2 bit informasi.

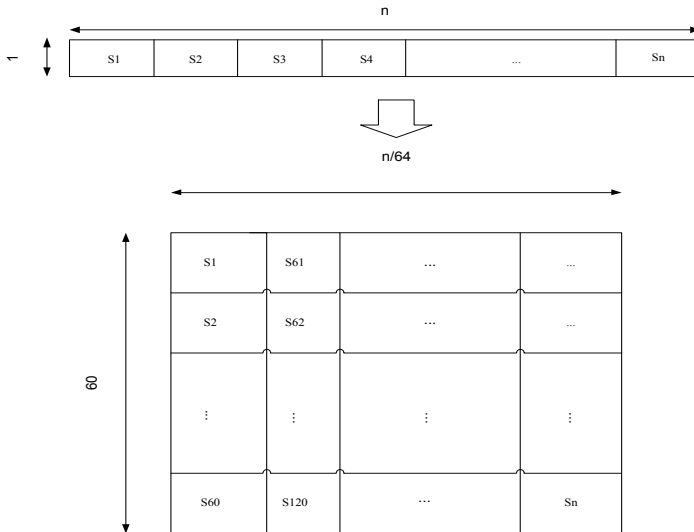


Gambar 3-3. Konstelasi *mapping* simbol QPSK $\theta = \pi/2$

Tabel 3-3. Nilai simbol *mapping* dari informasi

Nilai biner	Nilai desimal	Simbol	Nilai Simbol
00	0	S_1	$0.7071 + 0.7071j$
01	1	S_2	$-0.7071 + 0.7071j$
10	2	S_3	$0.7071 - 0.7071j$
11	3	S_4	$-0.7071 - 0.7071j$

Kemudian dari setiap deretan simbol – simbol *mapping* akan disusun paralel sesuai dengan banyaknya jumlah sub-carrier pada sistem OFDM. Ukuran paralel sebesar = (jumlah *subcarrier* – jumlah *pilot*) x (jumlah simbol *mapping* / jumlah *subcarrier*). Ilustrasi proses pengubahan bentuk serial menjadi bentuk paralel diperlihatkan pada



Gambar 3-4. Ilustrasi konversi *serial to parallel*

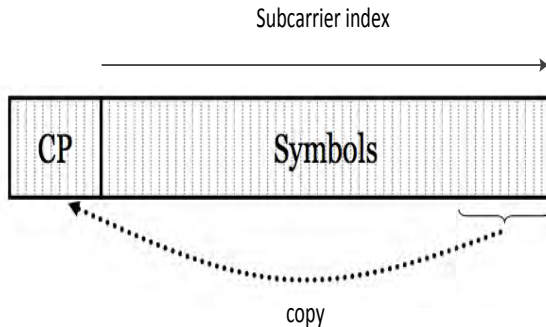
1.1.1 Penyisipan *pilot*

Susunan *pilot* pada sistem yang akan dibangun menggunakan tipe *comb*. *Pilot* diletakkan pada *subcarrier* 8, 22, 44 dan 58. Sedangkan sisa dari *subcarrier* yang ada digunakan sebagai tempat dari informasi.

1.1.2 IFFT dan Penambahan CP

sesudah ditambahkan *pilot* dan berbentuk paralel. Maka dari masing –masing *subcarrier* diaplikasikan pada FFT sehingga masing – masing *subcarrier* menghasilkan sinyal dalam domain waktu.

setelah data dan *pilot* ditempatkan pada posisi masing – masing maka langkah selanjutnya adalah menambahkan *cyclic prefix* dengan ukuran 25% dari satu simbol OFDM. Dalam hal ini, satu simbol OFDM memiliki ukuran panjang 64 sehingga ukuran *cyclic prefix* sebesar 16. Salinan dari *cyclic prefix* kemudian diambil dari posisi *subcarrier* ke-49 hingga ke-64, yang kemudian ditempatkan pada bagian depan seperti terlihat pada **Gambar 3-5**.



Gambar 3-5. Ilustrasi penyalinan *cyclic prefix*

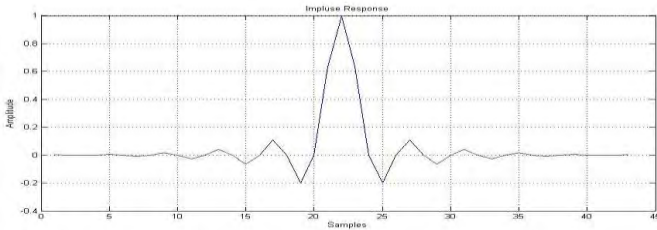
1.1.3 Parallel to serial dan penambahan preamble

Kemudian setelah penambahan *cyclic prefix* langkah selanjutnya adalah mengubah urutan data dari paralel menjadi susunan serial. Sehingga susunan serial dari satu baris *subcarrier* beserta *cyclic prefix* merupakan satu simbol OFDM. Maka luaran dari *parallel to serial* ini adalah susunan serial simbol simbol OFDM.

Dari susunan serial simbol – simbol OFDM tersebut sebelum ke tahap selanjutnya maka ditambahkan simbol – simbol berupa *preamble* yang berfungsi nantinya sebagai proses korelasi. Proses korelasi ini bertujuan untuk 2 hal yaitu melakukan estimasi kanal dan juga menentukan awal dari *frame* OFDM yang pertama.

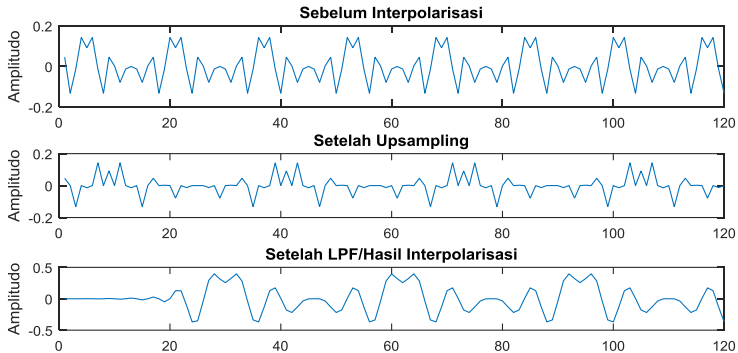
Preamble disini merupakan gabungan antara *Short Training Symbol* (STS) dan *Long Training Symbol* (LTS), susunan dari *preamble* ini diperlihatkan padagambar dibawah.

Pada Gambar 3-7 diatas dapat terlihat bahwa pada proses interpolasi terdapat dua tahap. Tahap pertama yaitu *Up-Sampling* dengan ukuran L yaitu $L > 1$ dan tahapan terakhir yaitu sinyal hasil dari proses interpolasi itu akan masuk kedalam *Low Pass Filter* (LPF). Dimana LPF ini merupakan response impulse dari $h[m]$. LPF digunakan untuk memfilter sinyal dari hasil *up-sample*. Pada tugas akhir ini, L yang digunakan yaitu 2, yang harus dipahami yaitu semakin besar ukuran dari L maka akan semakin sedikit data yang bisa dikirim dalam 1 *frame*. Sementara untuk respon impuls dari LPF yang sudah dinormalisasi yang digunakan pada tugas akhir ini diperlihatkan pada Gambar 3-9 dibawah ini.



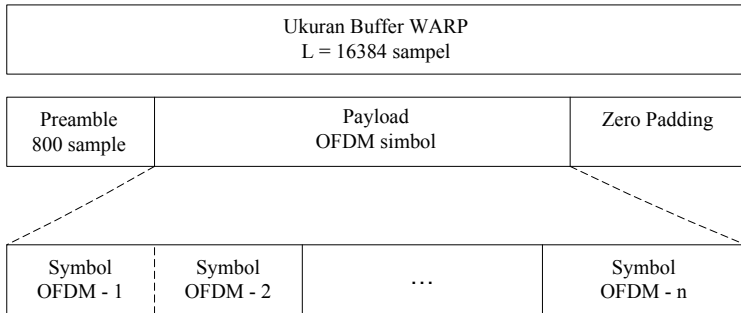
Gambar 3-8. Respon impuls LPF interpolarisasi

Setelah melalui proses interpolasi selesai maka sinyal akan mengalami langkah terakhir sebelum dipancarkan yaitu sinyal akan dinormalisasi untuk mengoptimalkan DAC yang terdapat pada WARP, sehingga nilai sinyal akan berada pada *range* +1 dan -1. Hasil dari interpolarisasi dapat digambarkan pada **Gambar 3-9**.



Gambar 3-9. Gambaran Output dari setiap blok interpolarisasi

Setelah proses interpolasi selesai, maka secara keseluruhan, desain *frame* komunikasi dapat digambarkan pada **Gambar 3-10**.



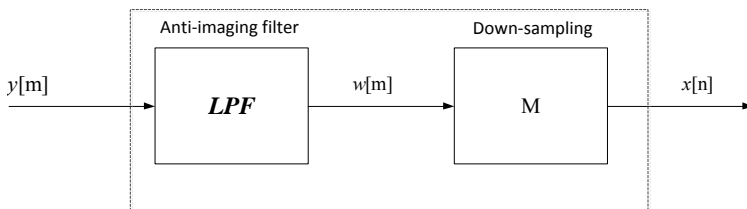
Gambar 3-10. Struktur frame sistem OFDM

3.3 Perancangan Sistem Penerima OFDM

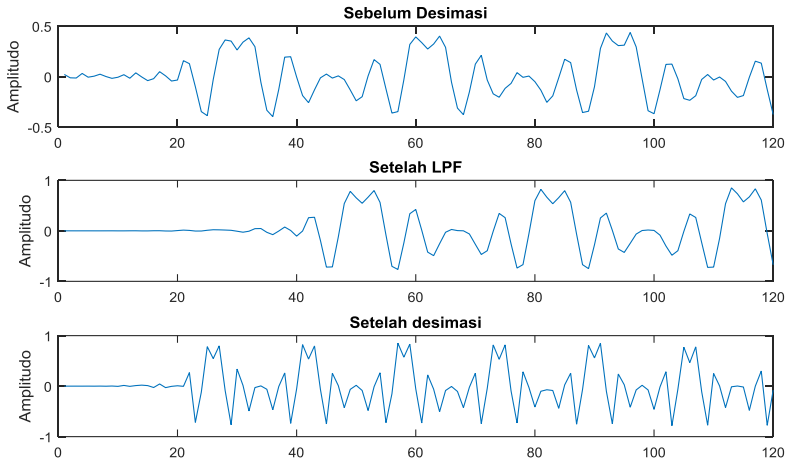
Pada bagian penerima OFDM ini terdiri dari beberapa bagian seperti yang sudah dijelaskan sebelumnya pada bab 2 mengenai bagian – bagian yang menyusun sistem OFDM. Diantaranya adalah desimasi, korelasi LTS, estimasi kanal, ekstraksi payload, *serial to parallel*, korelasi kesalahan fase, *demapping* dan terakhir *parallel to serial*.

3.3.1 Desimasi

Pada proses desimasi ini perbedaan yang mencolok yaitu jika pada proses interpolasi ada blok diagram *up-sampling* dengan variable L maka proses desimasi ini terdapat blok diagram *down-sampling* dengan variable M . Namun yang harus diingat yaitu nilai L dan M haruslah sama. Hasil dari interpolasi dapat digambarkan pada **Gambar 3-12**.



Gambar 3-11. Blok diagram desimasi



Gambar 3-12. Gambaran Output dari setiap blok desimasi

3.3.2 Korelasi LTS

Setelah sinyal mengalami proses desimasi, langkah selanjutnya yaitu melakukan *cross correlation* antara *preamble* yang terdapat pada penerima dengan satu LTS yang terdapat pada bagian pemancar. Rumus untuk *cross correlation* digambarkan pada persamaan (3.1):

$$C(n) = \sum_{l=0}^M \sum_{k=1}^N r(l*N + k + n)s^*(l*N + k) \quad (3.1)$$

Dimana:

- r : Keseluruhan *preamble* pada penerima
- s : Satu buah LTS
- N : Panjang dari data OFDM (64 sample)
- M : Jumlah dari LTS yang di *cross correlation*

Proses korelasi ini bertujuan untuk 2 hal yaitu melakukan estimasi kanal dan juga menentukan awal dari *frame* OFDM yang pertama. Untuk proses estimasi kanal sendiri akan dibahas lebih mendalam pada sub-bab selanjutnya. Proses korelasi ini terjadi antara sinyal yang diterima dengan dengan deretan 1 buah LTS yang terdapat pada pemancar sehingga akan didapatkan dua nilai puncak hasil

korelasi dengan LTS dikarenakan pada *preamble* terdapat dua buah LTS dengan syarat 0.8 dari nilai maksimum. Nilai puncak yang terakhir dari hasil korelasi tersebut yang akan digunakan untuk mendapatkan *frame* OFDM yang pertama.

3.3.3 Estimasi kanal

Kemudian setelah mendapatkan hasil korelasi maka akan mendapatkan *frame* awal OFDM. Estimasi kanal digunakan untuk ekualisasi setiap symbol OFDM didalam sinyal terima. Nilai estimasi kanal digunakan untuk mengkompensasi dari pengurangan nilai magnitude yang disebabkan oleh frekuensi selektif dari kanal *multipath* dan *noise*. Proses estimasi kanal ini menggunakan metode estimasi kanal *Least-Square* (LS).

Sebuah LTS pada penerima yang sudah melalui proses blok FFT dinotasikan dengan $R_{l,k}$ ($l = 0,1$) digambarkan dalam bentuk dimana $W_{l,k}$ ditambah dengan hasil perkalian dari LTS pada pemancar $d_{l,k}$ dan kanal $H_{l,k}$.

$$R_{l,k} = H_{l,k}d_{l,k} + W_{l,k} \quad (3.2)$$

Persamaan estimasi kanal menggunakan LTS berdasarkan [22] diperlihatkan pada persamaan 3.3. Setelah nilai dari estimasi kanal didapat, maka dilakukan proses ekualisasi terhadap *payload* (total keseluruhan symbol OFDM) yang diterima dengan menggunakan persamaan [17] seperti yang diperlihatkan pada persamaan 3.4.

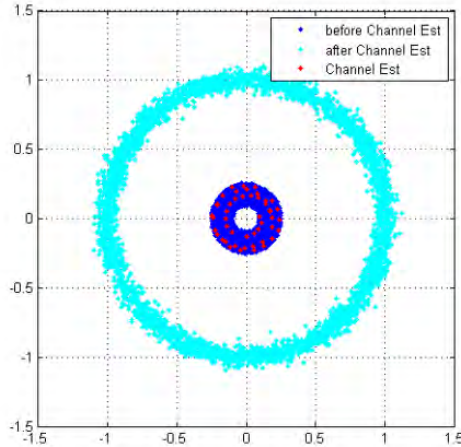
$$\begin{aligned} \widehat{H}_k &= \frac{1}{2}(R_{0,k} + R_{1,k}) \frac{1}{d_k} \\ &= \frac{1}{2}(H_k d_k + W_{0,k} + H_k d_k + W_{1,k}) \frac{1}{d_k} \\ &= H_k + \frac{1}{d_k} \frac{(W_{0,k} + W_{1,k})}{2} \end{aligned} \quad (3.3)$$

Pada prakteknya, perubahan nilai saat sebelum melalui estimasi dan sesudah estimasi dalam bentuk konstelasi sangat jelas terlihat seperti yang ditunjukkan pada **Gambar 3-13**. Jika dilihat pada Gambar tersebut, fungsi dari estimasi kanal ini yaitu untuk mengembalikan bentuk sinyal ke lingkaran asalnya.

$$xe(k) = x(k) / \widehat{H}_k \quad (3.4)$$

Dimana

- $xe(k)$: Sinyal setelah melalui proses ekualisasi
- $x(k)$: Sinyal sebelum melalui proses ekualisasi
- \widehat{H}_k : Nilai estimasi kanal



Gambar 3-13. Konstelasi pada proses estimasi kanal

3.3.4 Koreksi kesalahan fase

Setelah proses estimasi kanal selesai, maka langkah selanjutnya yaitu melakukan koreksi fasa error. Estimasi ini diperlukan karna meskipun sinyal sudah berada pada lingkaran yang tepat namun masih terdapat sisa (*residual*) error pada *timing offset* yang disebabkan oleh adanya karakteristik variasi waktu pada kanal. Dampak dari error ini yaitu ketidaktepatan letak symbol-symbol OFDM [15].

Seperti yang diperlihatkan pada Gambar 3.6 terdapat 4 buah *pilot* yang terdapat pada masing-masing symbol OFDM, *pilot-pilot* tersebut yang akan digunakan untuk mengurangi sisa error pada *timing offset* tersebut. Penempatan letak *pilot* tersebut bertujuan untuk menghindari *pilot* yang akan dikirim terganggu satu sama lain [15].

Proses estimasi fasa error cukup simple yaitu dengan mengalikan symbol yang diterima dengan $e^{-j\theta}$ dimana θ adalah sudut

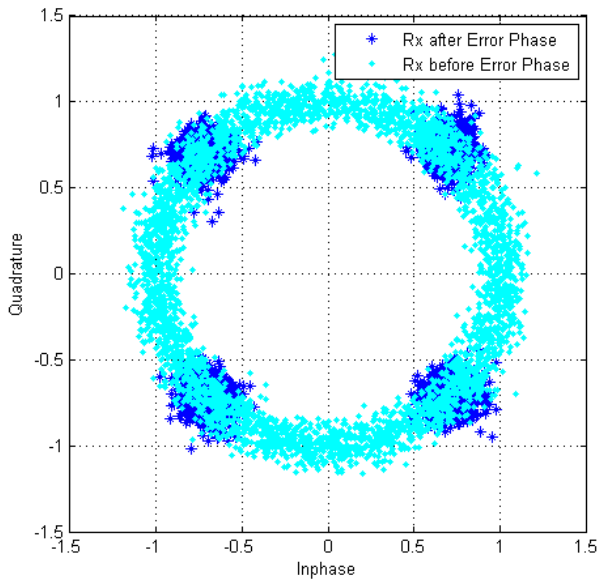
dari estimasi fasa. Secara matematis, operasi perbaikan fasa dilakukan dengan persamaan sebagai berikut:

$$s'(k) = s(k)e^{-j\theta} \quad (3.5)$$

Dimana:

- $s'(k)$: Nilai sinyal diterima setelah proses koreksi fasa
- $s(k)$: Nilai sinyal diterima sebelum proses koreksi fasa
- θ : Sudut dari estimasi fasa

Nilai θ didapat dari rata-rata perbedan nilai sudut antara *pilot* yang diterima dengan *pilot* yang dipancarkan. Efek dari penggunaan koreksi fasa error pada proses perbaikan fasa bisa dilihat pada **Gambar 3-14**.



Gambar 3-14. Perbandingan Fasa pada Sinyal dengan Koreksi Fasa

3.3.5 Ekstraksi payload dan *serial to parallel*

Langkah selanjutnya adalah membuang *cyclic prefix* dan memisahkan *pilot* dan payload informasi yang ada pada frame. membuang *cycle prefix* pada frame terletak pada baris pertama hingga baris ke-16, sisanya adalah payload beserta *pilot*. setelah membuang *cyclic prefix*, maka kemudian *pilot* dipisahkan dengan cara menyisihkan data yang terletak pada indeks baris *pilot*, yaitu baris ke 8, 22, 44 dan

58. Setelah itu data yang susunanya berupa data parallel, maka diubah menjadi susunan data berurut serial.

3.3.6 Demapping QPSK dan parallel to serial

Proses *demapping* merupakan proses kebalikan dari proses *mapping*. Data yang berupa nilai pada domain sumbu *in-phase* dan *quadrature* diubah menjadi nilai – nilai bit penyusunnya. Setiap satu simbol *mapping* akan diubah menjadi 2 bit . terlihat pada merupakan nilai – nilai dari setiap simbol.

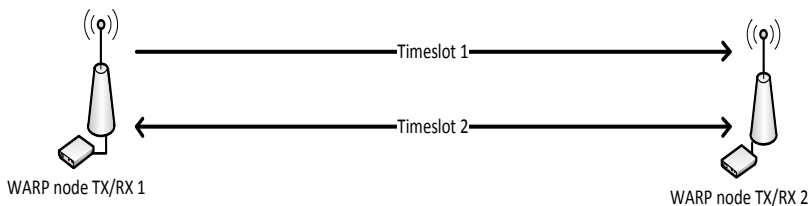
Tabel 3-4. Nilai simbol *demapping*

Nilai biner	Nilai Simbol	Nilai desimal	Simbol
00	$0.7071 + j0.7071$	0	S_1
01	$-0.7071 + j0.7071$	1	S_2
10	$0.7071 - j0.7071$	2	S_3
11	$-0.7071 - j0.7071$	3	S_4

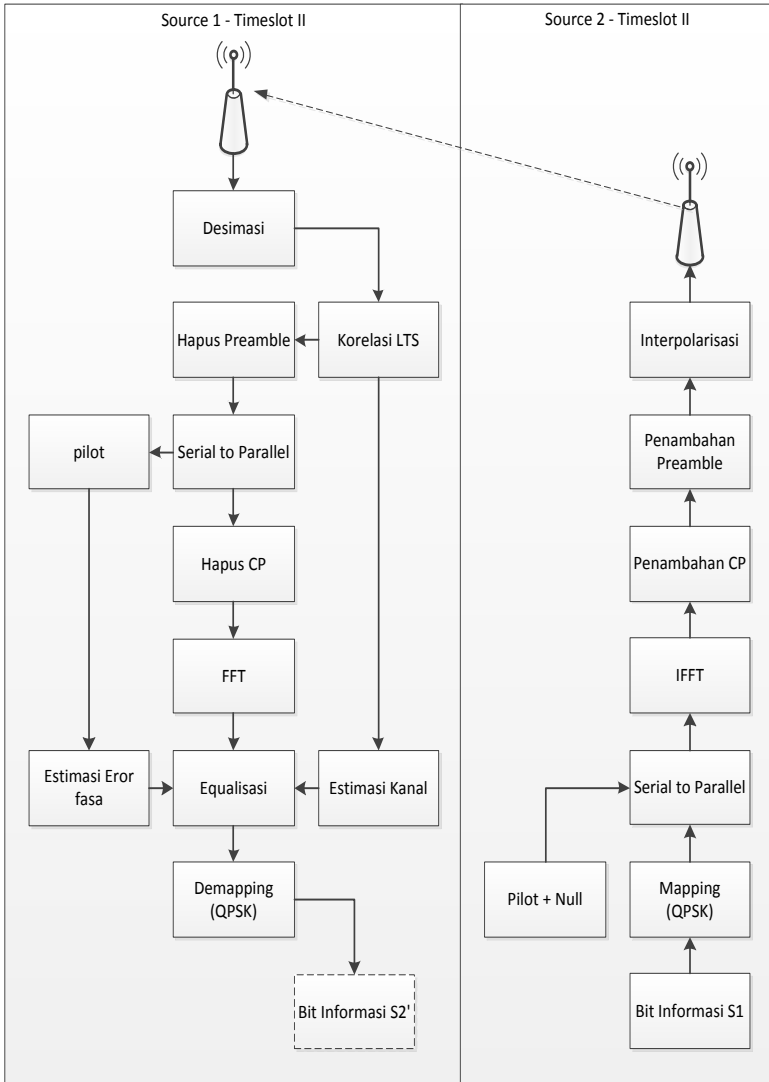
3.4 Perancangan Sistem Komunikasi OFDM *Direct*

Pada perancangan sistem komunikasi OFDM *direct* perangkat WARP yang digunakan berjumlah dua buah, seperti yang diperlihatkan dari. Setiap *node* berfungsi sebagai pengirim dan penerima.

Pada Error! Reference source not found. memperlihatkan tahapan proses pengiriman dan penerimaan, pada *timeslot* pertama S1 mengirimkan ke *node* S2 kemudian terakhir pada **Gambar 3-17** diperlihatkan *timeslot* ke-dua *node* 2 mengirimkan informasi ke *node* S1.



Gambar 3-15. Skema komunikasi OFDM *direct*

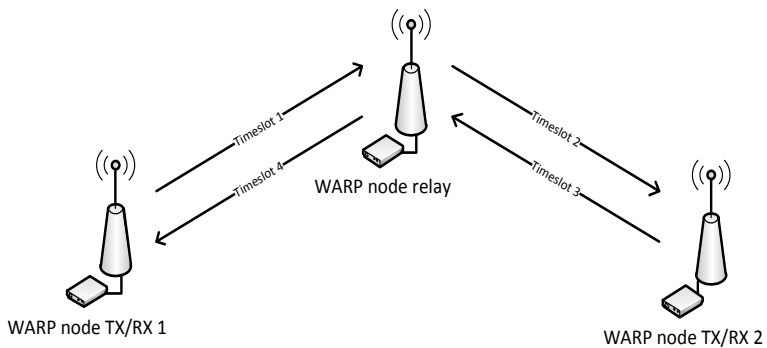


Gambar 3-17. Proses pengiriman dan penerima skema *direct* pada timeslot II

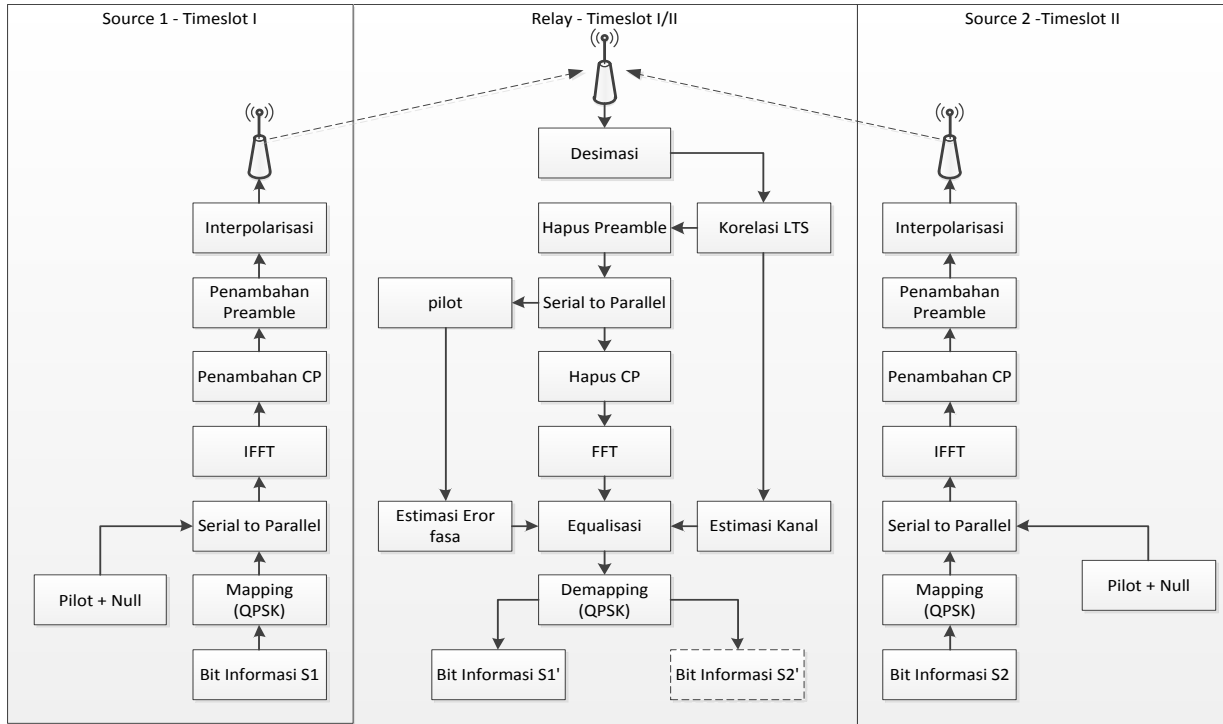
3.5 Perancangan sistem komunikasi OFDM *multihop* konvensional

Sistem konvensional *multihop* menggunakan skema seperti yang ditunjukkan pada **Gambar 18**. Skema ini menggunakan perangkat WARB sejumlah tiga buah, dua diantaranya digunakan sebagai *node* pengirim/penerima dan satu *node* digunakan sebagai *node relay*. Tahap dari skema ini untuk melakukan satu *cycle* pengiriman dan penerimaan membutuhkan empat *timeslot* seperti yang diperlihatkan pada **Gambar 19**.

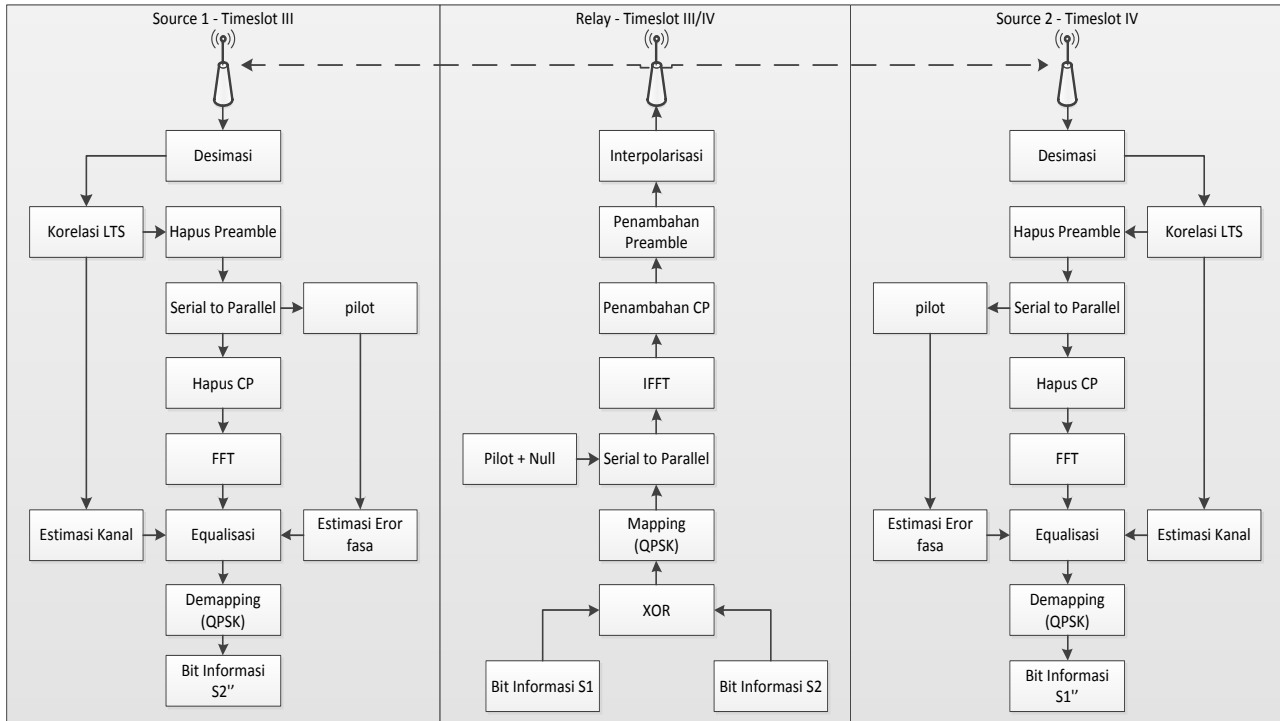
Pada *timeslot* pertama *node* S1 mengirimkan data ke *node relay*, *node* S1 dikonfigurasi sebagai pengirim dan *node relay* sebagai penerima. Selanjutnya di *timeslot* ke-dua *node relay* meneruskan data yang telah diterima dari S1 menuju *node* S2, dengan kata lain *node relay* dikonfigurasi sebagai pengirim dan *node* S2 sebagai penerima. Kemudian pada *timeslot* ke-tiga *node* S2 dikonfigurasi sebagai pengirim dan *node relay* sebagai penerima, *node* S2 mengirimkan data ke *node relay*. Kemudian pada *timeslot* terakhir *node relay* meneruskan data kepada *node* 2, *node relay* mengirimkan data ke *node* S1, *node relay* dikonfigurasi sebagai pengirim dan *node* S1 sebagai penerima. Proses pengiriman dan penerimaan tersebut diperlihatkan pada Error! Reference source not found. dan **Gambar 3-20**.



Gambar 3-18. Skema komunikasi OFDM *multihop* konvensional



Gambar 3-19. Proses pengiriman dan penerima skema *multihop* pada timeslot I dan II

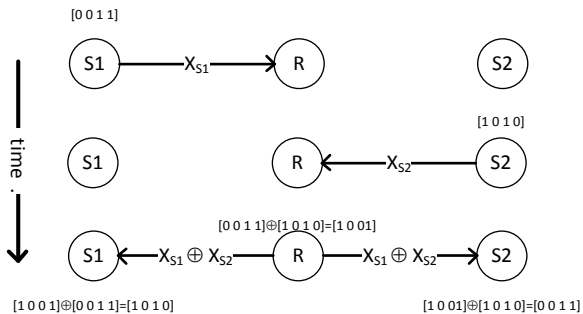


Gambar 3-20. Proses pengiriman dan penerima skema *multihop* pada timeslot III dan IV

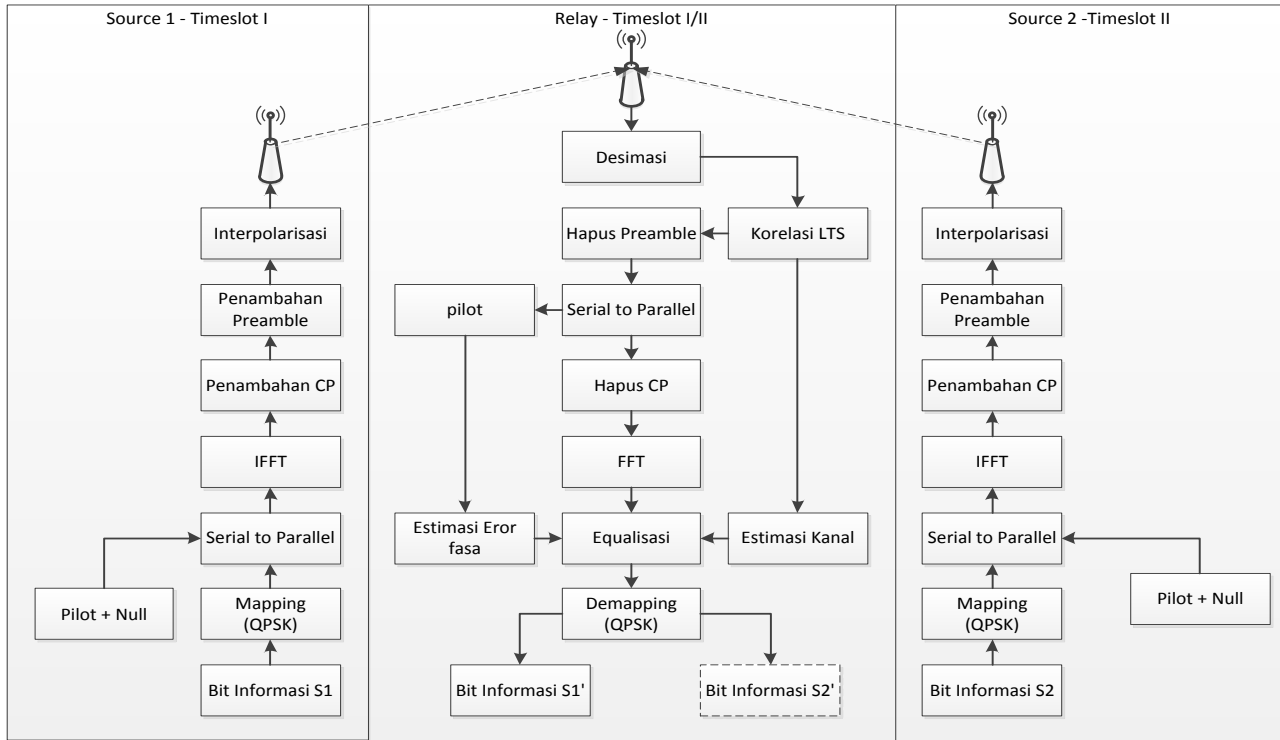
3.6 Perancangan sistem komunikasi OFDM dengan *network coding*

Sistem konvensional *multihop* menggunakan skema empat buah *timeslot*, sedangkan dengan skema *network coding*, sistem hanya menggunakan tiga buah *timeslot* untuk melakukan satu *cycle* proses pengiriman dan penerimaan. Skema ini menggunakan perangkat WARB sejumlah tiga buah, dua diantaranya digunakan sebagai *node* pengirim/penerima dan satu *node* digunakan sebagai *node relay*. Tahap dari skema ini untuk melakukan satu *cycle* pengiriman dan penerimaan diperlihatkan pada **Gambar 3-21**.

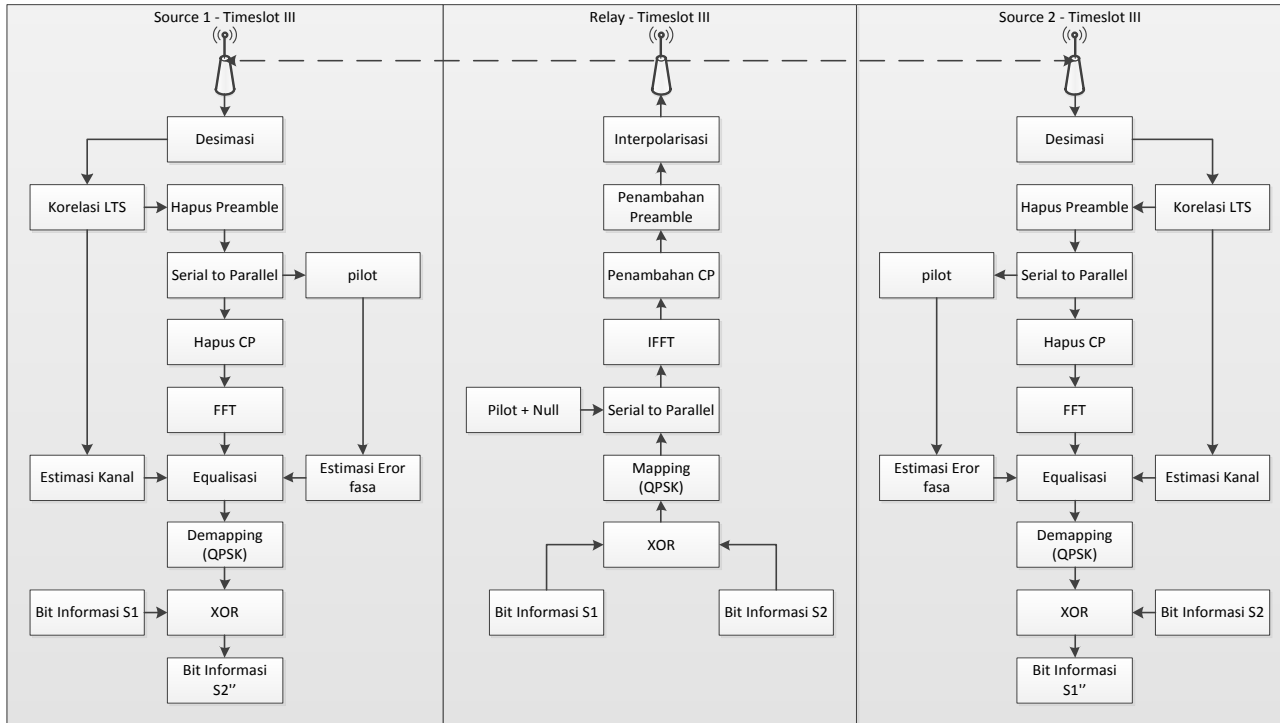
Pada **Gambar 3-22** dan **Gambar 3-23** dapat dijelaskan bahwa *timeslot* pertama *node* S1 mengirimkan data ke *node relay*, *node* S1 dikonfigurasi sebagai pengirim dan *node relay* sebagai penerima. Selanjutnya pada *timeslot* ke-dua *node* S2 dikonfigurasi sebagai pengirim dan *node relay* sebagai penerima, *node* S2 mengirimkan data ke *node relay*. yang menjadi pembeda konfigurasi antara skema *multihop* konvensional dengan skema *network coding* adalah pada *timeslot* ke-tiga, yaitu pada *timeslot* ke-tiga *node relay* melakukan fungsi tambahan yaitu mengkombinasikan dua informasi yang diterima dari S1 dan S2, pengkombinasian yang dimaksud dalam bentuk XOR dari bit – bit informasi yang diterima dari kedua sumber. Pada *timeslot* terakhir ini, setelah *node relay* melakukan fungsi *network coding*, *node relay* meneruskan data kepada *node 2* dan *node 3*, *node relay* dikonfigurasi sebagai pengirim sedangkan *node* S1 dan *node* S2 sebagai penerima. Pada penerima, data yang masing – masing telah didapatkan selanjutnya dilakukan XOR kembali dengan data yang dikirimkan oleh *node* masing – masing, sehingga data yang dikirimkan dari masing – masing pengirim dapat diterjemahkan.



Gambar 3-21. Skema komunikasi OFDM *network coding*



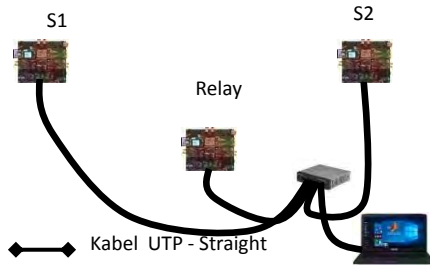
Gambar 3-22. Proses pengiriman dan penerima skema network coding pada timeslot I dan II



Gambar 3-23. Proses pengiriman dan penerima skema *network coding* pada timeslot III dan IV

3.7 Integrasi PC/Laptop ke perangkat WARP

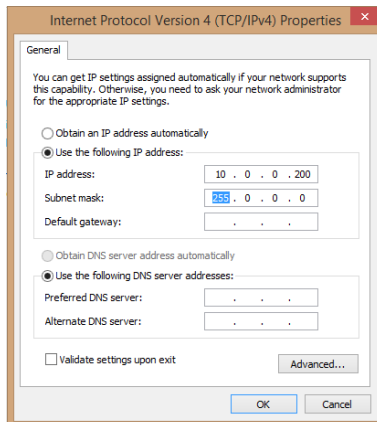
Pada penelitian ini digunakan tiga buah *node* WARP dengan konfigurasi *node* 1 dan 3 sebagai *node source* dan *destination* sedangkan *node* 2 dikonfigurasi sebagai *node relay*. WARP dengan PC/Laptop dihubungkan dengan menggunakan *switch Ethernet* dan kabel LAN seperti dapat dilihat pada **Gambar 3-24**



Gambar 3-24. Integrasi PC/Laptop dengan Modul WARP

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

1. Menggabungkan modul WARP dan PC/Laptop menggunakan *switch Ethernet* dan kabel LAN seperti pada gambar. Mengatur IP Lokal yang ada pada PC/Laptop seperti pada **Gambar 3-25**.



Gambar 3-25. Pengaturan IP Address pada PC/Laptop

- Mengatur IP *address* pada WARP, karena WARP yang digunakan adalah tiga buah, maka IP yang dipakai, 10.0.0.1 untuk *node* 1, 10.0.0.2 untuk *node* 2, 10.0.0.3 untuk *node* 3, pengaturan dilakukan dengan mengubah *dipswitch* yang ada pada modul WARP, seperti pada **Gambar 3-26**



Gambar 3-26. Konfigurasi *dipswitch* Untuk Mengatur IP pada WARP

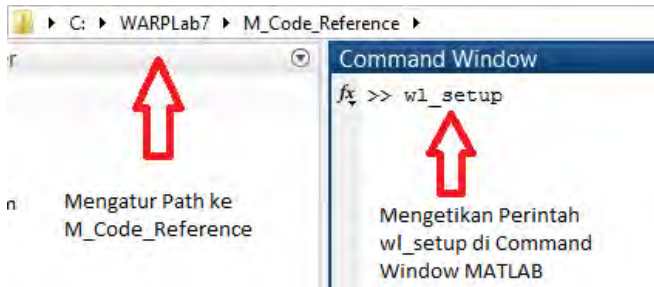
- Memastikan setiap *node* sudah terhubung ke *ethernet switch* dengan cara tes koneksi ping ke setiap *node*
ping 10.0.0.1 ke node 1
ping 10.0.0.2 ke node 2
ping 10.0.0.3 ke node 3
- Mengisi tabel routing lewat *command prompt* dengan mengetikkan perintah “arp -s 10.0.0.255 ff-ff-ff-ff-ff-ff”, kemudian memeriksa tabel routing dengan mengetikkan “arp -a”



Gambar 3-27. Pengaturan Tabel Routing

- Mengatur pengaturan WARP pada MATLAB dengan cara, mengarahkan direktori MATLAB ke “M-Code Reference”

WARPLab 7, kemudian ketikkan perintah “wl_setup” pada *command window* MATLAB



Gambar 3-28. Set Path Folder WARPLab pada MATLAB

- Memastikan koneksi modul WARP dengan PC/Laptop dengan mengetikkan perintah “wl_initNodes(3)” pada *command window*, angka dalam kurung menyatakan bahwa ada 3 *nodes* yang digunakan, untuk indikator keberhasilannya dapat dilihat pada **Gambar 3-29**.

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

>>

Gambar 3-29. Konfigurasi *Nodes* pada MATLAB

3.8 Skenario Pengukuran

Untuk mengukur beberapa parameter kinerja dari skema *network coding* pada sistem komunikasi OFDM, maka perlu dilakukan beberapa skenario pengukuran. Beberapa keadaan maupun skenario pengukuran dilakukan agar hasil yang didapatkan memberikan informasi yang lebih komprehensif.

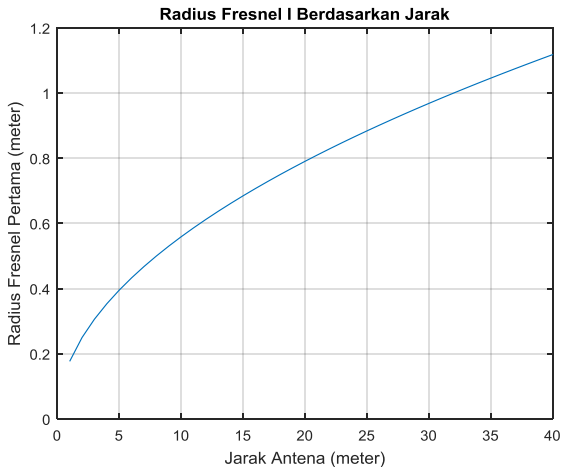
Adapun dari ruang lingkup fisik pengukuran dilakukan di dalam dan di luar ruangan, pengukuran seperti ini dilakukan guna memberikan gambaran mengenai unjuk kerja dari skema *network coding* pada kanal yang memiliki banyak pantulan dari sinyal yang dipancarkan dan kanal yang relatif memiliki sedikit pantulan. Kemudian skenario pengukuran selanjutnya adalah dengan dan tidak menggunakan penghalang pada jalur langsung antara pengirim dan penerima, skenario ini dilakukan pada kondisi pengukuran dalam ruangan

Pada kondisi menggunakan penghalang, penghalang dikondisikan sedemikian rupa agar jalur langsung dari pengirim dan penerima terdapat obstacle. Dalam hal ini memperhitungkan jari – jari fresnel pertama dari pengirim dan penerima, wilayah fresnel pertama ini akan akan dihalangi oleh penghalang yang telah dibuat. Untuk mempermudah memperlihatkan jari – jari fresnel dalam setiap pengukuran, maka dapat diperhitungkan menggunakan **persaman (3.6)**. pada **Gambar 3-29** memperlihatkan secara mudah hasil dari perhitungan **persamaan (3.6)**.

$$F_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \tag{3.6}$$

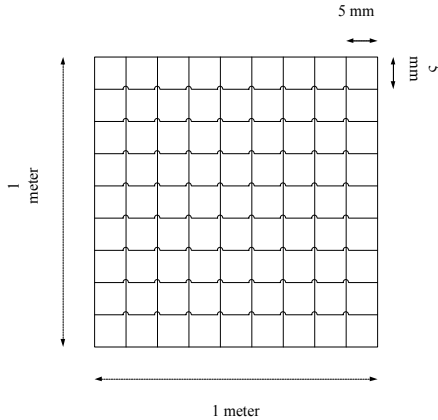
Dimana :

- F : Jari jari fresnel
- λ : panjang gelombang
- d1 : jarak dari pengirim ke titik jari – jari F
- d2 : jarak dari penerima ke titik jari – jari F



Gambar 3-30. Grafik radius fresnel pertama fungsi dari jarak antena

Kemudian setelah dimensi dari fresnel diketahui maka dimensi dari penghalang yang akan digunakan dapat dibuat. Pada pengukuran ini menggunakan penghalang dengan dimensi 1x1 meter dapat dilihat seperti pada **Gambar 3-31**.



Gambar 3-31. Dimensi dari penghalang

Selain digunakan untuk membuat skenario penghalang grafik pada Gambar 3-30 tentu juga digunakan sebagai referensi dari tinggi antenna yang akan digunakan. Tinggi antenna yang akan dibuat tentunya menyesuaikan agar lintasan propagasi antara pengirim/penerima ke *relay* dalam keadaan tanpa penghalang.

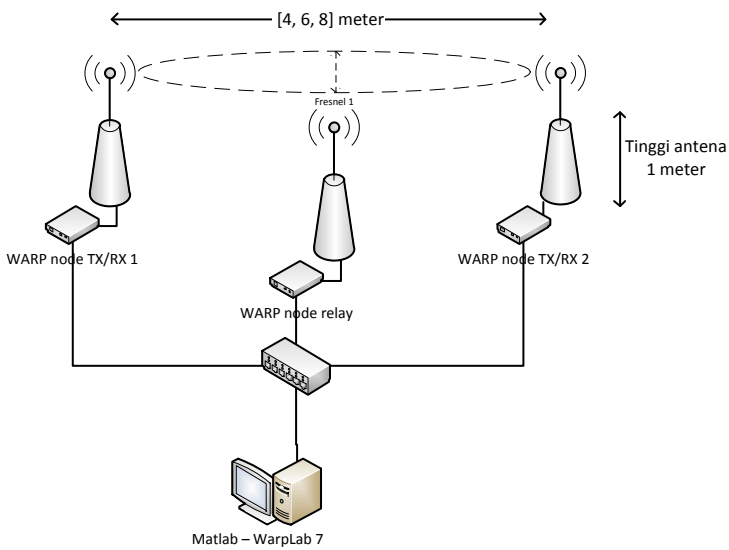
Pada akhirnya proses pengukuran untuk membandingkan unjuk kerja sistem *network coding* sistem komunikasi OFDM adalah sebagai berikut :

1. Pengukuran kondisi tanpa penghalang di dalam ruangan dengan variasi berbagai besaran jarak dan daya pancar
2. Pengukuran kondisi dengan penghalang di dalam ruangan dengan variasi berbagai besaran jarak dan daya pancar
3. Pengukuran kondisi tanpa penghalang di luar ruangan dengan variasi berbagai besaran jarak dan daya pancar
4. Pengukuran kondisi dengan penghalang di luar ruangan dengan variasi berbagai besaran jarak dan daya pancar

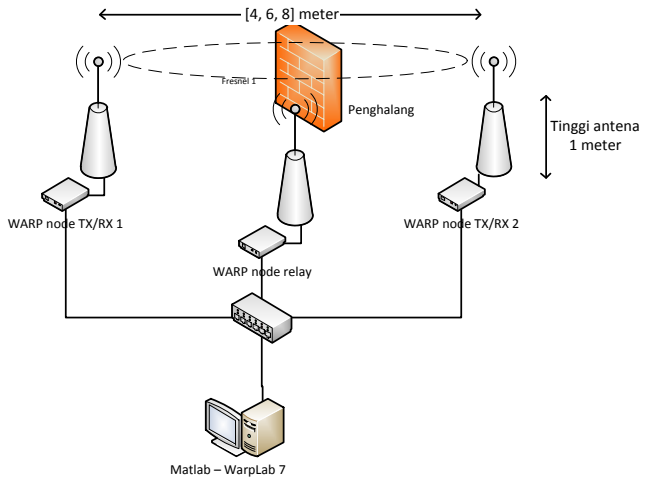
3.8.1 Pengukuran lingkungan dalam ruangan

Untuk lingkungan *indoor* pengukuran dilakukan didalam lab B304 dengan jarak antar *node* pengirim dan penerima 4, 6 dan 8 m dengan *relay* tepat berada ditengah – tengah *node* pengirim dan penerima. Masing-masing jarak tersebut memiliki dua kondisi yang berbeda yaitu tanpa penghalang seperti pada **Gambar 3-32** dan dengan

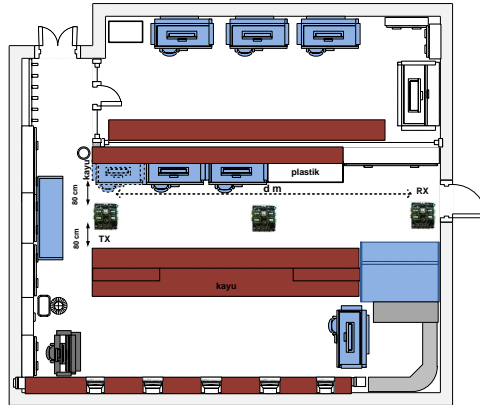
penghalang **Gambar 3-33**. Sketsa untuk pengukuran untuk lingkungan *indoor* dengan kondisi ruangan yang diperlihatkan pada **Gambar 3-34**.



Gambar 3-32. Pengaturan posisi pengukuran tanpa penghalang



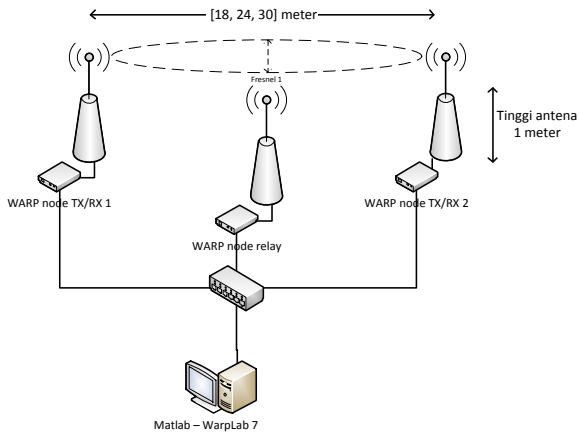
Gambar 3-33. Pengaturan posisi pengukuran dengan penghalang



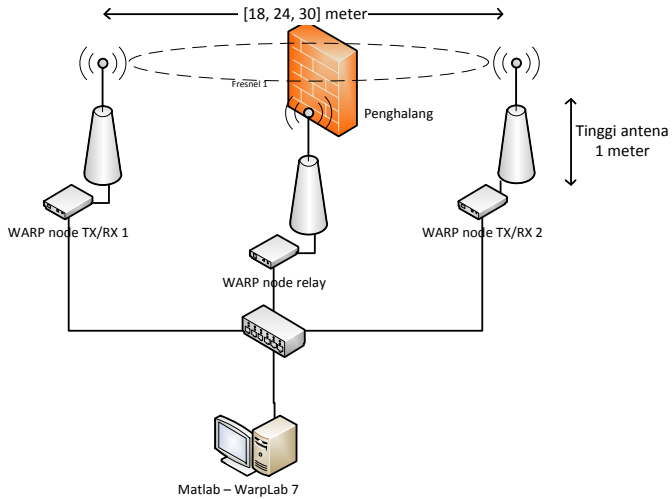
Gambar 3-34. Gambaran pengukuran lingkungan di dalam ruangan

1.1.4 Pengukuran lingkungan luar ruangan

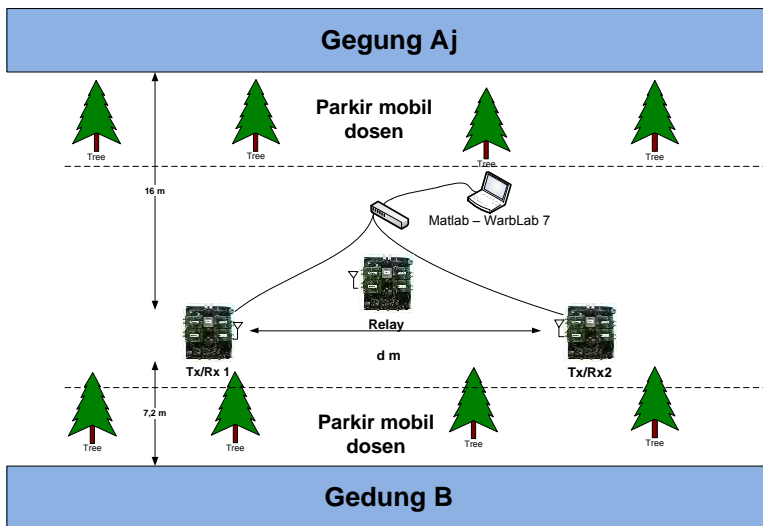
Untuk lingkungan *outdoor* pengukuran dilakukan di halaman parkir jurusan Elektro ITS dengan jarak antar *node* pengirim dan penerima 18, 24 dan 30 m dengan *relay* tepat berada ditengah – tengah *node* pengirim dan penerima. Masing-masing jarak tersebut memiliki dua kondisi yang berbeda yaitu tanpa penghalang seperti pada **Gambar 3-35** dan dengan penghalang **Gambar 3-36**. Sketsa untuk pengukuran untuk lingkungan *indoor* dengan kondisi ruangan yang diperlihatkan pada **Gambar 3-37**



Gambar 3-35. Pengaturan posisi pengukuran tanpa penghalang



Gambar 3-36. Pengaturan posisi pengukuran dengan penghalang



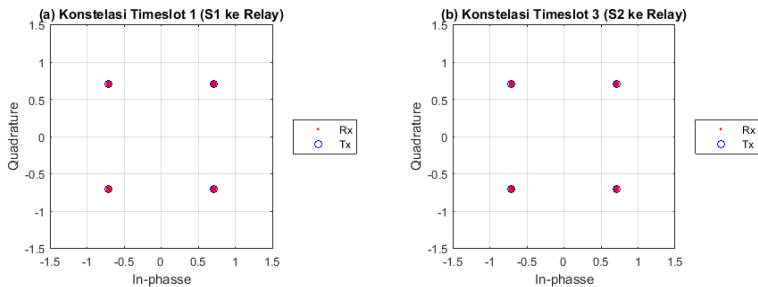
Gambar 3-37. Gambaran pengukuran lingkungan di luar ruangan

3.9 Validasi Menggunakan Simulasi Pada Kanal Ideal (*Noiseless*)

Pada alur perancangan sistem yang diperlihatkan pada **Gambar 3-1** memperlihatkan tahap pertama yang dilakukan dalam proses implementasi dari sistem ini adalah memastikan bahwa sistem telah berjalan sesuai dengan sebagaimana mestinya. Indikatornya adalah apabila sistem dijalankan pada simulasi menggunakan kondisi ideal atau bisa dikatakan tanpa *noise* maka semua data yang dikirimkan akan diterima keseluruhan dengan benar ($P_e=0$).

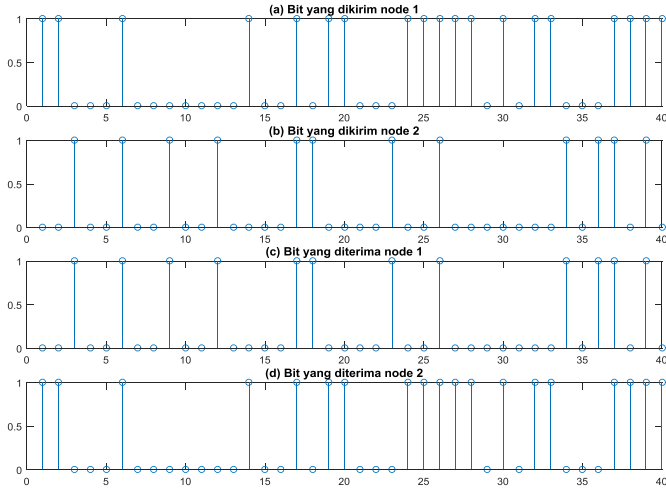
Simulasi dilakukan padatnya skema penelitian, skema *direct*, skema *multihop* konvensional dan skema *network coding*. Hasil yang diperlihatkan pada validasi ini adalah konstelasi, dimana konstelasi ini memperlihatkan benar bahwa pada simulasi ini dikondisikan pada kanal *noiseless*. Selanjutnya adalah memperlihatkan perbandingan dari bit informasi yang dikirimkan dan yang diterima di masing – masing node.

3.9.1 Skema *direct*



Gambar 3-38. Konstelasi simulasi skema *direct* pada kondisi kanal ideal (*noiseless*)

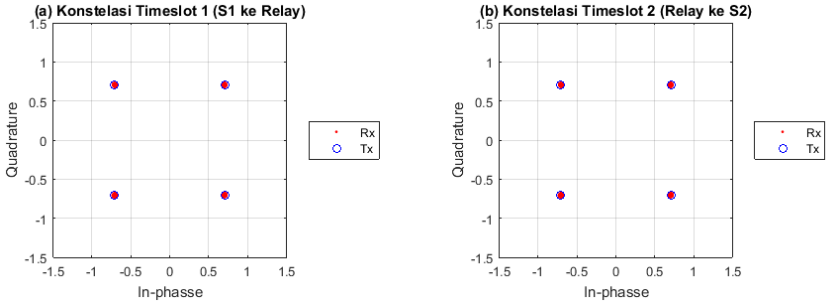
Telihat dari **Gambar 3-38** (a) dan (b) bahwa hasil simulasi dari pengiriman ke node S1 dari S2 maupun sebaliknya diterima tanpa adanya *noise*. Dengan ini maka proses pada penerima jika semua proses pada penerima berjalan dengan baik maka seluruh bit diterima tanpa adanya bit yang *error*.



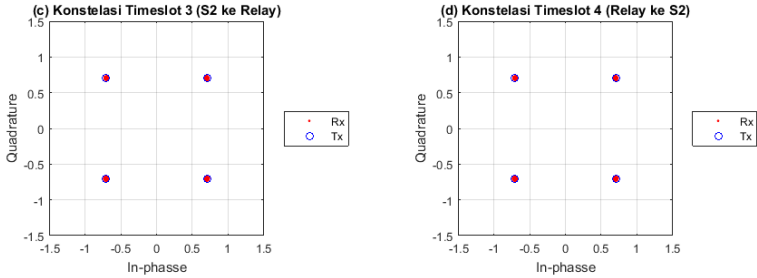
Gambar 3-39 Cuplikan bit informasi simulasi skema *direct* pada kondisi kanal ideal (*noiseless*)

Lalu setelah data *in-phase* dan *quadrature* yang diterima tanpa terpengaruh *noise*, selanjutnya diperlihatkan **Gambar 3-39** merupakan hasil dari pengirim dan penerima OFDM pada skema *direct* telah berjalan dengan baik (tidak terdapat eror), sehingga tahapan selanjutnya untuk bisa diimplementasikan secara nyata dapat dilakukan.

3.9.2 Skema *multihop* konvensional

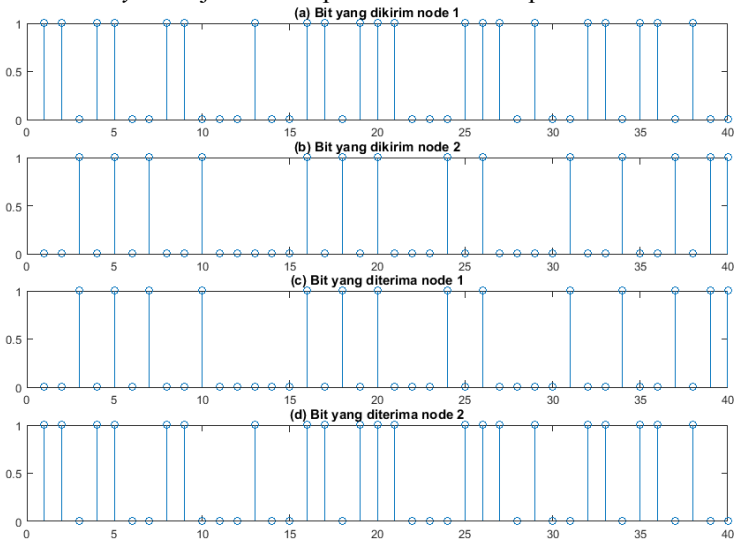


Gambar 3-40. Konstelasi simulasi skema *multihop* pada kondisi kanal ideal (*noiseless*) I



Gambar 3-41. Konstelasi simulasi skema *multihop* pada kondisi kanal ideal (*noiseless*) II

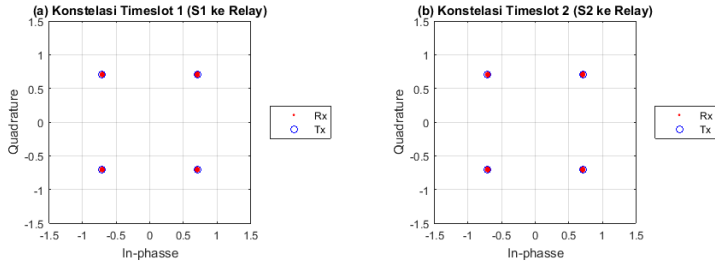
Telihat dari **Gambar 3-40 (a)** bahwa konstelasi dari simbol yang dikirim dari S1 ke node relay diberikan tanpa *noise*. **(b)** merupakan pengiriman dari *node relay* menuju node S2 pada *timeslot* II tanpa noise. Selanjutnya **(c)** merupakan pengiriman dari *node S2* menuju *node relay* pada *timeslot* III tanpa *noise*. Dan terakhir **(d)** merupakan pengiriman dari *node relay* menuju node S2 pada *timeslot* II tanpa noise.



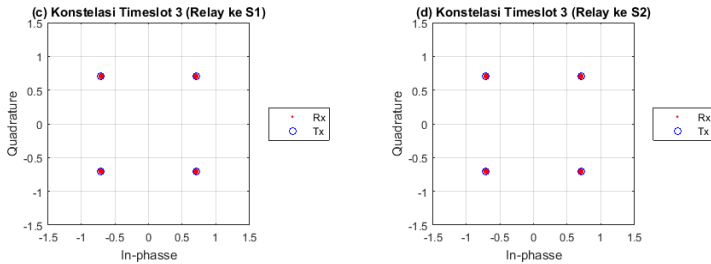
Gambar 3-42. Cuplikan bit informasi simulasi skema *multihop* pada kondisi kanal ideal (*noiseless*)

Selanjutnya **Gambar 3-42** menunjukkan hasil dari pengirim dan penerima OFDM pada skema *multihop* telah berjalan dengan baik (tidak terdapat error), sehingga tahapan selanjutnya untuk bisa diimplementasikan secara nyata dapat dilakukan.

3.9.3 skema network coding

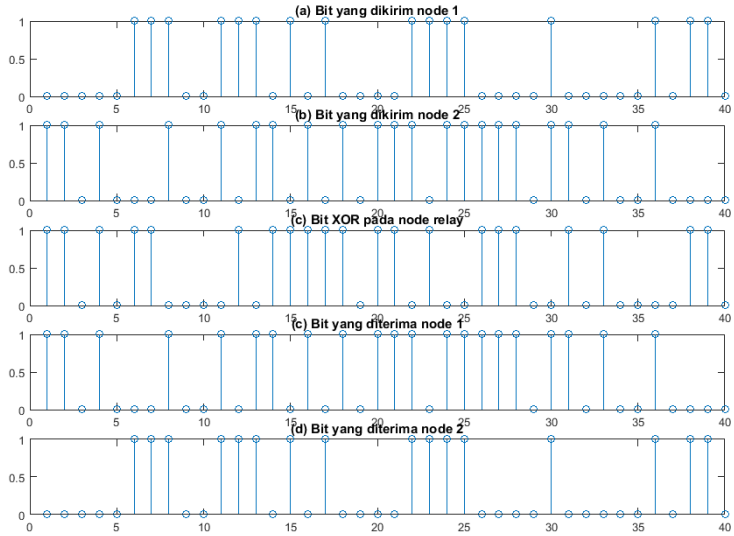


Gambar 3-43. Konstelasi simulasi skema *network coding* pada kondisi kanal ideal (*noiseless*) I



Gambar 3-44. Konstelasi simulasi skema *network coding* pada kondisi kanal ideal (*noiseless*) II

Telihat dari Gambar 3-44 (a) bahwa konstelasi dari simbol yang dikirim dari S1 ke node relay diberikan tanpa *noise*. (b) merupakan pengiriman dari *node* S2 menuju node *relay* pada *timeslot* II tanpa *noise*. Selanjutnya (c) dan (d) merupakan data XOR dari node S1 dan S2 yang *broadcast* dari *node relay* menuju *node* S1 dan S2 pada *timeslot* III tanpa *noise*.



Gambar 3-45. Cuplikan bit informasi simulasi skema *network coding* pada kondisi kanal ideal (*noiseless*)

Pada **Gambar 3-45** terlihat bit - bit yang ditransmisikan baik dari S1 dan S2, juga pada relay berupa bit XOR dari bit –bit S1 dan S2, gambar tersebut juga menunjukkan hasil dari pengirim dan penerima OFDM pada skema *network coding* telah berjalan dengan baik (tidak terdapat error), sehingga tahapan selanjutnya untuk bisa diimplementasikan secara nyata dapat dilakukan.

Halaman ini sengaja dikosongkan

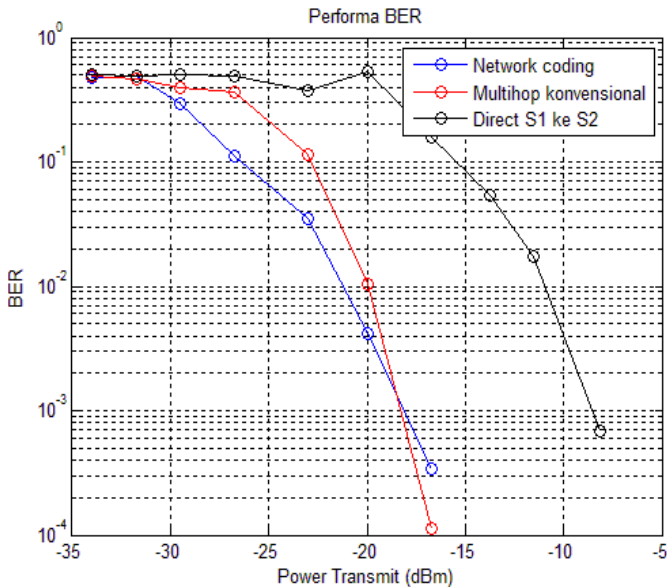
BAB 4 HASIL PENGUKURAN DAN ANALISA

Pada bab ini hasil dari simulasi dan implementasi sistem pada WARP diperlihatkan dan kemudian dianalisa. Hasil simulasi dan Pengukuran menghasilkan beberapa parameter yang diukur meliputi *bit error rate*, waktu satu *cycle* transmisi dan memperlihatkan gambar hasil transmisi baik itu skema komunikasi *direct*, *multihop* konvensional dan *network coding*.

4.1 Hasil dan Analisa Pengukuran BER

Pada pengukuran ini dilakukan pengamatan dari kinerja sistem *direct*, *multihop* maupun *network coding*. Parameter yang diamati adalah *probability of error* atau biasa disebut *bit error rate* (BER). Pengukuran dilakukan pada luar maupun dalam ruangan, yang bertujuan untuk mengetahui pengaruh fenomena *multipath*. Selanjutnya pengukuran dilakukan pada jarak yang berbeda - beda, sehingga nantinya diharapkan dapat mengetahui jarak optimum dari sistem.

4.1.1 Pengukuran BER pada kondisi tanpa penghalang

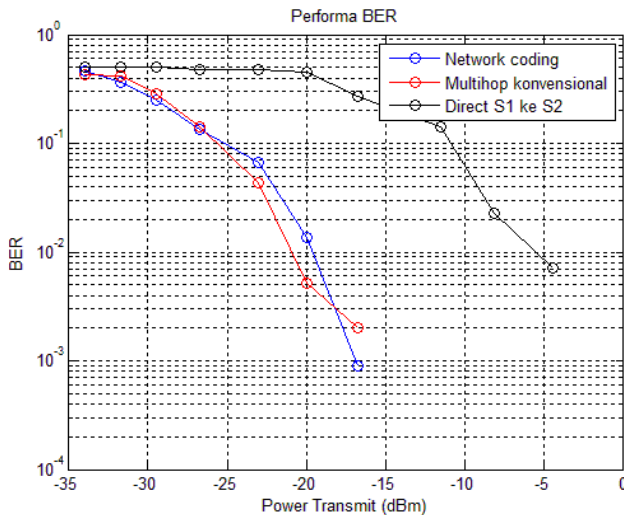


Gambar 4-1. Kinerja BER pada kondisi tanpa penghalang

Pengukuran BER dilakukan pada masing – masing skema .Perhitungan BER dilakukan dengan membandingkan informasi yang diterima lalu menghitung rasio antara jumlah bit yang salah dengan total keseluruhan bit. Pada **Gambar 4-1** memperlihatkan tiga buah grafik BER fungsi dari daya pancar. Data didapat dari hasil pengukuran di dalam ruangan dengan kondisi tanpa penghalang dengan jarak antara S1 dan S2 sejauh 4 meter. Terlihat bahwa performa dari skema *multihop* dan *network coding* lebih baik dibandingkan dengan skema *direct*. Untuk mencapai nilai BER bernilai 10^{-2} pada skema *direct* dibutuhkan daya pancar sebesar ± -11 dBm, sedangkan masing masing pada skema *network coding* dan *multihop* hanya membutuhkan daya pancar sebesar ± -20 dBm. Sehingga penggunaan skema *network coding* maupun *multihop* memiliki *gain* relatif sebesar ± -9 dB dari daya yang dibutuhkan skema *direct* untuk mendapatkan nilai BER sebesar 10^{-2} . Namun pada skema *network coding* dan *multihop* perlu penambahan berupa *resource* yaitu penambahan *node relay*

4.1.2 Pengukuran BER pada kondisi dengan penghalang

Parameter yang diukur sama dengan kondisi di dalam ruangan pada kondisi LOS, perbedaanya adalah topologi pengukuran yang diperlihatkan kan **Gambar 3-33** di bab 3 hasil yang didapat adalah sebagai berikut



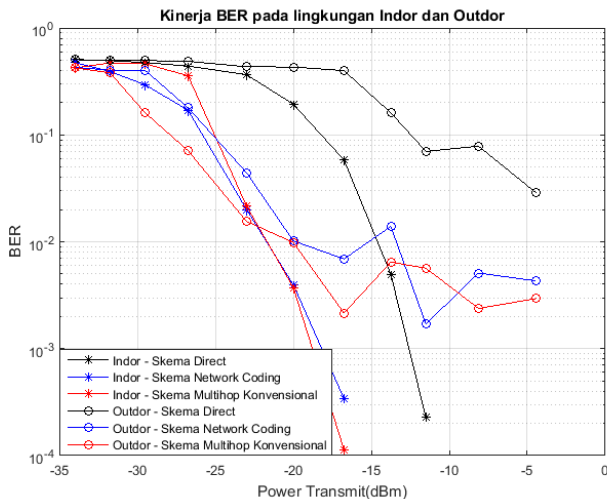
Gambar 4-2. Kinerja BER pada kondisi dengan penghalang

Sama pada analisa pada **Gambar 4-2** memperlihatkan tiga buah grafik BER fungsi dari daya pancar. Data didapat dari hasil pengukuran di dalam ruangan dengan kondisi LOS dengan jarak antara S1 dan S2 sejauh 4 meter. Terlihat bahwa performa dari skema *multihop* dan *network coding* lebih baik dibandingkan dengan skema *direct*. Untuk mencapai nilai BER bernilai 10^{-2} pada skema *direct* dibutuhkan daya pancar sebesar $\pm -6\text{dBm}$, sedangkan masing masing pada skema *network coding* dan *multihop* hanya membutuhkan daya pancar sebesar $\pm -20\text{dBm}$. Sehingga penggunaan skema *network coding* maupun *multihop* memiliki *gain* relatif sebesar $\pm -14\text{dB}$ dari daya yang dibutuhkan skema *direct* untuk mendapatkan nilai BER sebesar 10^{-2} .

Terlihat bahwa perbedaan ketika menggunakan penghalang pada komunikasi *direct*. Pada *multihop* dan *network coding* relatif tidak berpengaruh, dikarenakan penghalang yang diberikan diletakkan pada jalur langsung antara S1 dan S2. Pada kondisi tanpa penghalang didapat nilai BER bernilai 10^{-2} pada skema *direct* dibutuhkan daya pancar sebesar $\pm -11\text{dBm}$ sedangkan dengan penghalang dibutuhkan daya sebesar $\pm -6\text{dBm}$, jadi penghalang berpengaruh sekitar 5dB pada jarak 4 meter.

4.1.3 Pengukuran BER pada lingkungan *outdoor* dan *indoor*

Pada pengukuran ini akan diperlihatkan unjuk kerja dari BER untuk setiap skema. Hasil yang didapat adalah sebagai berikut :

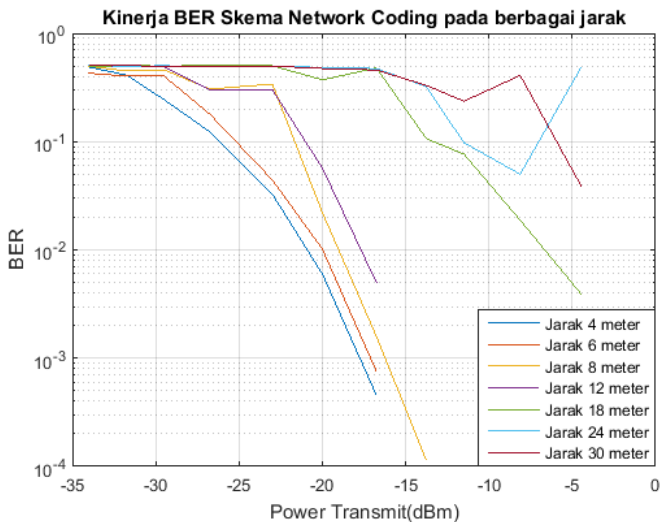


Gambar 4-3. Kinerja BER pad lingkungan *indoor* dan *outdoor*

Pada **Gambar 4-3** memperlihatkan hasil dari pengukuran ketiga skema pada lingkungan *indoor* dan *outdoor*. Terlihat hasil pengukuran pada lingkungan *indoor* memperlihatkan hasil pengukuran yang lebih baik. Ketika daya pancar -20dBm , skema *network coding* di lingkungan *indoor* memiliki nilai BER $\pm 3 \times 10^{-3}$ sedangkan *outdoor* bernilai $\pm 10^{-2}$. Skema *multihop* memperlihatkan hasil yang sama jika diamati pada daya pancar -20dBm , di lingkungan *indoor* memiliki nilai BER $\pm 3 \times 10^{-3}$ sedangkan *outdoor* bernilai $\pm 10^{-2}$. Dan terakhir skema *direct* di lingkungan *indoor* memiliki nilai BER $\pm 1 \times 10^{-1}$ sedangkan *outdoor* bernilai $\pm 3 \times 10^{-1}$. Dari ketiga skema memperlihatkan bahwa hasil pengukuran pada kondisi *indoor* lebih baik. Beberapa faktor yang dapat mempengaruhi diantaranya, pada pengukuran lingkungan *outdoor* daya tak langsung memberikan efek yang tidak terlalu besar, namun terdapat banyak interferensi dari perangkat – perangkat pemancar Wifi yang frekuensi kerjanya sama dengan sistem yang di bangun.

4.1.4 Pengukuran BER skema *network coding* pada berbagai jarak

Selanjutnya pengukuran dilakukan adalah dengan skenario jarak yang berbeda - beda, sehingga nantinya diharapkan dapat mengetahui jarak optimum sistem. Pengukuran dilakukan pada jarak 4 meter, 6 meter, 8 meter, 12 meter, 18 meter, 24 meter dan 30 meter. Hasil dari pengukuran diperlihatkan pada gambar berikut :

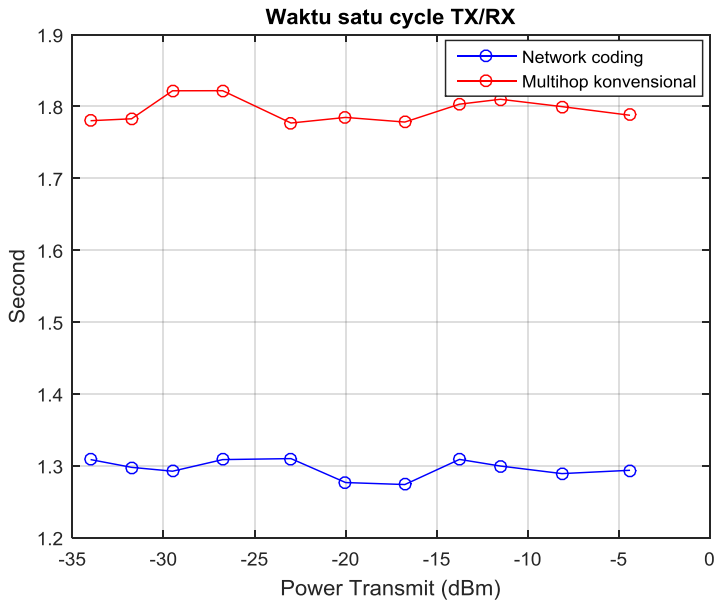


Gambar 4-4. Kinerja BER pada berbagai Jarak

Hasil dari pengukuran pada berbagai jarak memperlihatkan bahwa untuk mendapatkan kinerja BER senilai 10^{-3} pada daya pancar dibawah -15dBm maka jarak yang dapat mencapainya adalah ketika jarak dibawah 12 meter. Pada jarak 18, 24 dan 30 meter nilai BERnya terukur masih lebih besar dari 10^{-1} .

4.2 Hasil dan Analisa Pengukuran Waktu Komunikasi

Pada pengukuran BER di **sub-bab 4.2** dilakukan pengujian dari skema *direct*, *multihop* dan *network coding* ditemukan performa BER skema *multihop* dan *network coding* lebih baik dari skema *direct*, sehingga untuk memperlihatkan performa manakah yang lebih baik dari skema *multihop* atau *network coding*, maka dilakukan pengukuran terhadap waktu transmisi dari masing – masing skema. Pengukuran waktu komunikasi dilakukan pada masing – masing skema komunikasi *multihop* dan *network coding*. Waktu yang diukur adalah waktu sepanjang satu siklus pengiriman dan penerimaan informasi dari S1 dan S2. Selanjutnya hasil diperlihatkan dari gambar dibawah ini.



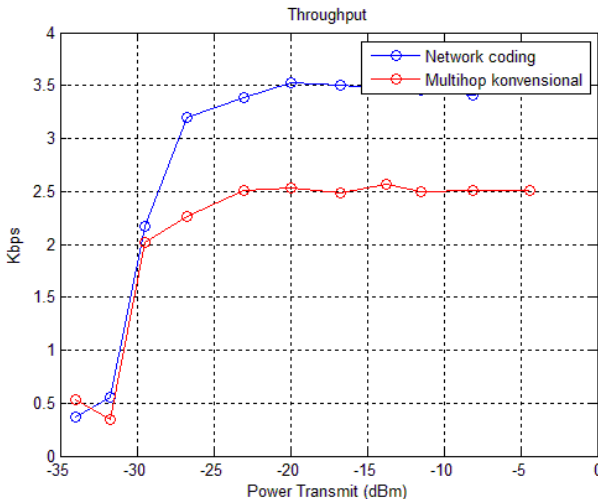
Gambar 4-5. Waktu satu kali transmisi

Terlihat di **Gambar 4-5** adalah dua buah grafik dari hasil pengukuran waktu transmisi dari beberapa kali pengiriman dengan daya pancar yang berbeda beda. Waktu transmisi *network coding* (grafik berwarna biru) terlihat lebih cepat daripada waktu transmisi dari skema *multihop* (grafik warna merah). Masing masing bernilai rata – rata 1.3 detik dan 1.8 detik. Artinya skema *network coding* lebih cepat 0.5 detik atau 27 % dari skema *multihop*.

Peningkatan ini dikarenakan pada skema multihop membutuhkan 4 *timeslot* untuk melakukan *transmit* dan *recieve*, sedangkan pada *network coding* membutuhkan hanya 3 *timeslot* untuk melakukan *transmit* dan *recieve*. Ini merupakan bentuk efisiensi yang dilakukan penggunaan skema *network coding*.

4.3 Hasil dan Analisa Pengukuran *Throughput*

Dari hasil pengamatan terhadap parameter BER dan waktu transmisi maka dapat dilakukan pendekatan untuk memperhitungkan nilai *throughput*. Pendekatan untuk mencari nilai *throughput* didapat dengan cara mencari jumlah *successfull* bit yang dikirim pada satu siklusnya lalu dibagi dengan total waktu satu siklus transmisi. Maka hasil yang didapatkan diperlihatkan oleh



Gambar 4-6. *Throughput* dari skema *network coding* dan *multihop*

Pada Gambar 4-6 memperlihatkan bahwa *throughput* mulai relatif stabil pada daya pada *transmitter* bernilai lebih dari -20dBm. Dan

terlihat perbedaan dari nilai *throughput* maksimum berbeda antara skema *network coding* dan *multihop*. Skema *network coding* menghasilkan *throughput* maksimum 3.5 Kbps dan skema *multihop* sebesar 2.5Kbps. Skema *network coding* mampu memberikan improvisasi terhadap *throughput* lebih besar $\pm 28\%$ dari *throughput* maksimum skema *multihop*

4.4 Hasil dan Analisa Pengiriman Gambar

Pada pengujian selanjutnya memperlihatkan hasil dari pengiriman citra gambar dari masing masing skema. Dari hasil gambar yang diterima menggunakan ketiga skema tersebut, kemudian diperhitungkan MSE dan PSNR dari citra gambar tersebut.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE}$$

4.4.1 Pengukuran di dalam ruangan pada kondisi tanpa penghalang



(a) Citra Asli



MSE = 14.73
PSNR = 36.48 dB
(b) Citra *network Coding*



MSE = 20.03
PSNR = 35.15 dB
(c) Citra *Multihop*



MSE = 39.51
PSNR = 32.20 dB
(d) Citra *direct*

Gambar 4-7 . Hasil pengiriman gambar di dalam ruangan dengan kondisi tanpa penghalang

Pada gambar diatas terlihat bahwa hasil pengiriman gambar di dalam ruangan dengan kondisi tanpa penghalang . Hasil dari ketiga skema terlihat secara kasat mata bahwa kualitas gambar terlihat baik. Nilai PSNR menunjukkan bahwa kualitas gambar dari skema *network coding* lebih baik dengan nilai PSNR = 36.48 dB, kemudian dengan skema *multihop* dengan nilai PSNR = 35.15 dB pada urutan kedua dan terakhir dengan skema *direct* dengan PSNR 32.20 dB.

4.4.2 Pengukuran di dalam ruangan pada kondisi dengan penghalang



(a) Citra Asli



MSE = 627.71
PSNR = 20.19 dB
(b) Citra *network Coding*



MSE = 326.26
PSNR = 23.03 dB
(c) Citra *Multihop*



MSE = 3239.30
PSNR = 13.06 dB
(d) Citra *direct*

Gambar 4-8. Hasil pengiriman gambar di dalam ruangan dengan kondisi dengan penghalang

Gambar 4-8 memperlihatkan bahwa hasil pengiriman gambar di dalam ruangan dengan kondisi tanpa penghalang . Hasil dari ketiga skema terlihat secara kasat mata bahwa kualitas gambar terlihat baik. Namun terlihat pada gambar hasil pengiriman dengan skema *direct* terdapat beberapa titik *noise* yang tidak sesuai dengan gambar asli, hal ini menunjukkan ketika jalur langsung S1 Dan S2 terdapat penghalang hasil gambar yang diterima juga menunjukkan perubahan dan penggunaan skema *network coding* ataupun *multihop* dapat

memperbaiki performa komunikasi. Nilai PSNR menunjukkan bahwa kualitas gambar dari skema *multihop* lebih baik, kemudian dengan skema *network coding* pada urutan kedua dan terakhir dengan skema *direct*.

4.4.3 Pengukuran di luar ruangan pada kondisi tanpa penghalang



(a) Citra Asli



MSE = 627.71
PSNR = 20.19 dB

(b) Citra *network Coding*



MSE = 326.26
PSNR = 23.03 dB

(c) Citra *Multihop*



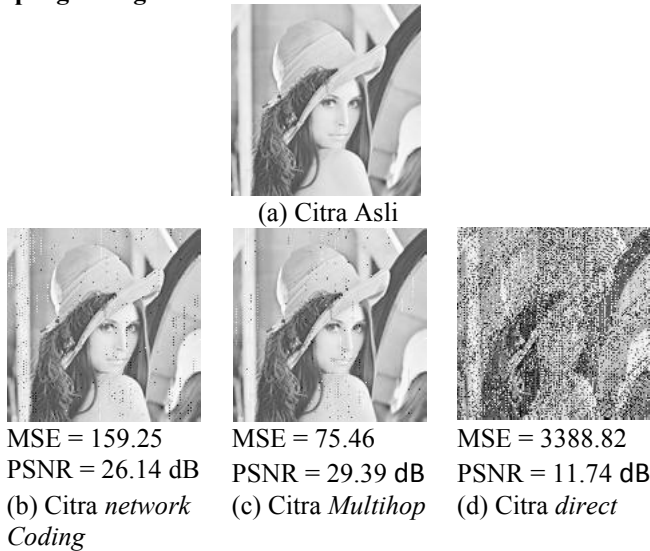
MSE = 3239.30
PSNR = 13.06 dB

(d) Citra *direct*

Gambar 4-9. Hasil pengiriman gambar di luar ruangan dengan kondisi tanpa penghalang

Pada kondisi pengukuran di luar ruangan, walaupun penggunaan skema *network coding* dan *multihop* memberikan perbaikan atas pengiriman skema *direct*, didapat bahwa *bit error* yang didapat terlihat lebih banyak. Beberapa hal yang menjadi perhatian yang diduga adalah faktor yang menyebabkan hal tersebut bisa terjadi adalah interferensi, ketika di luar ruangan terdapat banyak frekuensi yang menginterferensi sistem, dikarenakan WARP menggunakan frekuensi kerja pada band 2,4 GHz dan Wifi yang terdapat di area Gedung Elektro juga menggunakan frekuensi yang sama.

4.4.4 Pengukuran di luar ruangan pada kondisi dengan penghalang



Gambar 4-10. Hasil pengiriman gambar di luar ruangan dengan kondisi dengan penghalang

Pada kondisi pengukuran di luar ruangan menggunakan penghalang pada jalur langsung antara S1 dan S2, sama halnya pada pengukuran **sub-bab 4.4.3** walaupun penggunaan skema *network coding* dan *multihop* memberikan perbaikan atas pengiriman skema *direct*, didapat bahwa *bit error* yang didapat terlihat lebih banyak. Namun terlihat bahwa kondisi gambar yang dihasilkan bisa dikatakan sangat buruk. Ini juga membuktikan bahawa penggunaan skema *network coding* dapat melakukan perbaikan pada pengiriman gambar yang dilakukan.

Beberapa hal yang menjadi perhatian yang diduga adalah faktor yang menyebabkan hal tersebut bisa terjadi adalah interferensi, ketika di luar ruangan terdapat banyak frekuensi yang menginterferensi sistem, dikarenakan WARP menggunakan frekuensi kerja pada band 2,4 GHz dan Wifi yang terdapat di area Gedung Elektro juga menggunakan frekuensi yang sama.

BAB 5

PENUTUP

5.1 Kesimpulan

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

1. *Network coding* berkerja dengan memanfaatkan penghematan penggunaan *timeslot* komunikasi, dimana pada skema *multihop* konvensional menggunakan 4 *timeslot* sedangkan skema *network coding* menggunakan 3 *timeslot*. Pada skema *network coding* relay mengirimkan data berupa hasil *xor* dari kedua informasi sumber pada *timeslot* ketiga, kemudian pada penerima data di-*xor*-kan kembali dengan hasil yang dikirimkan *relay* dan data yang dikirimkan oleh node masing – masing ke node *relay* untuk mendapatkan informasi yang dikirimkan *node* lainnya.
2. Proses penggabungan informasi yang menggunakan mekanisme *xor* pada sistem komunikasi OFDM dilakukan pada bit – bit hasil demodulasi pada *node relay*, kemudian pada penerima untuk mendapatkan informasi yang dikirimkan juga dilakukan menggunakan mekanisme *xor* yang dilakukan pada bit – bit hasil demodulasi pada penerima/
3. Unjuk kerja skema *network coding* pada sistem komunikasi OFDM pada penelitian ini adalah sebagai berikut :
 - a. Penggunaan skema *network coding* dapat melakukan perbaikan kinerja BER dari skema *direct* terbukti pada pengukuran yang dilakukan diperlihatkan bahwa untuk mendapatkan nilai BER bernilai 10^{-2} pada skema *direct* dibutuhkan daya pancar sebesar ± -11 dBm, sedangkan pada skema *network coding* hanya membutuhkan daya pancar sebesar ± -20 dBm, namun pada skema *network coding* diperlukan penambahan *resource* berupa *node relay*.
 - b. Waktu transmisi pada skema *network coding* lebih cepat daripada waktu transmisi dari skema *multihop* Masing masing bernilai rata – rata 1.3 detik dan 1.8 detik. Artinya skema *network coding* lebih cepat 0.5 detik atau 27 % dari skema *multihop*
 - c. Penerapan skema *network coding* pada sistem komunikasi OFDM dapat meningkatkan *throughput* dikarenakan penransmisian dengan *network coding* dilakukan lebih cepat daripada menggunakan skema *multihop* konvensional,

tepatnya skema *network coding* pada tugas ini memiliki kinerja waktu pentransmisian $\pm 28\%$ lebih cepat daripada skema *multihop* tradisional.

5.2 Saran

Saran yang bisa diajukan sebagai bahan pengembangan atau penelitian lebih lanjut diantaranya :

1. Penelitian mengenai *multiuser* ($n > 2$) pada skema *network coding*.
2. Penelitian mengenai *physical network coding*
3. Penggunaan *error control coding* tertentu pada sistem, guna meningkatkan kinerja sehingga memiliki *probability of error* yang lebih baik
4. *Joint design* dari penelitian komunikasi kooperatif OFDM dan *network coding*
5. Penggunaan skema *physical network coding* guna meningkatkan performa dari waktu pentransmisian data

LAMPIRAN A : USULAN TUGAS AKHIR

Jurusan Teknik Elektro
Fakultas Teknologi Industri – ITS

TE 141599 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Bambang Eko Surya
Nomer Pokok : 2213106008
Bidang Studi : Teknik Telekomunikasi dan Multimedia
Tugas Diberikan : Semester Gasal 2015/2016
Dosen Pembimbing : 1. Dr. Ir. Suwadi, M.T.
2. Dr. Ir. Wirawan, DEA

16 SEP 2015


Judul Tugas Akhir : **Implementasi dan Analisa Kinerja Network Coding pada Sistem Komunikasi Orthogonal Frequency Division Multiplexing (OFDM) Menggunakan Wireless Open-Access Research Platform (WARP)**
Implementation and Performance Analysis of Network Coding on Orthogonal Frequency Division Multiplexing (OFDM) Using Wireless Open-Access Research Platform (WARP)

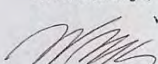
Uraian Tugas Akhir :

Network coding di dalam sistem komunikasi semakin populer dan berkembang dikarenakan kemampuannya dapat mengurangi *delay* dan meningkatkan *throughput* dan *robustness*. Disaat yang bersamaan, teknik modulasi *Orthogonal Frequency Division Multiplexing (OFDM)* juga semakin populer dan berkembang. Hadirnya OFDM memiliki keuntungan dalam mengefisienkan penggunaan spektrum *bandwidth* dan kecepatan transmisi yang tinggi. Sehingga nantinya penggabungan teknologi OFDM dan *network coding* dapat menggabungkan kelebihan dari masing – masing teknologi. Pada tugas akhir ini akan dilakukan simulasi kinerja *network coding* pada sistem komunikasi OFDM pada perangkat lunak Matlab dan implementasi pada *Wireless Open-Access Research Platform (WARP)*. Dengan menggunakan WARP kinerja antara sistem komunikasi OFDM dengan dan tanpa *network coding* dapat dibandingkan kemudian data tersebut akan dianalisa kinerjanya. Kinerja yang diukur diantaranya adalah besaran *Bit Error Rate (BER)* dan *throughput*.

Dosen Pembimbing I,

Dosen Pembimbing II,

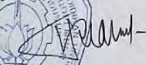

Dr. Ir. Suwadi, M.T.
NIP. 19680818 1993031 002

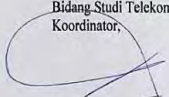

Dr. Ir. Wirawan, DEA
NIP. 19631109 1989031 011

Mengetahui,
Jurusan Teknik Elektro FTI-ITS
Ketua

Menyetujui,
Bidang Studi Telekomunikasi Multimedia
Kordinator




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


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

Halaman ini sengaja dikosongkan

LAMPIRAN B : PROGRAM MATLAB

`inisiiasi.m`

```
global USE_WARPLAB TXRX WRITE_PNG_FILES CHANNEL
N_OFDM_SYMS MOD_ORDER ...
    TX_SCALE INTERP_RATE SC_IND_PILOTS SC_IND_DATA
N_SC CP_LEN ...
    N_DATA_SYMS FFT_OFFSET LTS_CORR_THRESH
DO_APPLY_CFO_CORRECTION ...
    DO_APPLY_PHASE_ERR_CORRECTION
DO_APPLY_SFO_CORRECTION DECIMATE_RATE ...
    USE_AGC MAX_TX_LEN SAMP_PADDING CF rx_RSSI

% Params:
USE_WARPLAB_TXRX           = 1;
WRITE_PNG_FILES            = 0;
CHANNEL                     = 6;
CF                          = 0;

% Waveform params
N_OFDM_SYMS                = 92;
MOD_ORDER                  = 4;
TX_SCALE                   = 1.0;
INTERP_RATE                = 2;

% OFDM params
SC_IND_PILOTS              = [8 22 44 58];
SC_IND_DATA                = [2:7 9:21 23:27 39:43 45:57
59:64];
N_SC                       = 64;
CP_LEN                     = 16;
N_DATA_SYMS                = N_OFDM_SYMS *
length(SC_IND_DATA);      ;

% Rx processing params
FFT_OFFSET                 = 4;
LTS_CORR_THRESH            = 0.8;
DO_APPLY_CFO_CORRECTION   = 1;
DO_APPLY_PHASE_ERR_CORRECTION = 1;
DO_APPLY_SFO_CORRECTION   = 1;
DECIMATE_RATE              = INTERP_RATE;

% WARPLab experiment params
USE_AGC                    = 0;
MAX_TX_LEN                 = 2^19;
SAMP_PADDING               = 100;
```

main.m

```
%% INISIALISASI
inisialisasi;
BER_NC1 = []; BER_NC2 = [];
BER_TC1 = []; BER_TC2 = [];
BER_DR1 = []; BER_DR2 = [];
TR_NC = []; T_NC = [];
TR_TC = []; T_TC = [];
TR_DR = []; T_DR = [];
RSSI_NC1 = []; RSSI_NC2 = [];
RSSI_TC1 = []; RSSI_TC2 = [];
RSSI_DR1 = []; RSSI_DR2 = [];
Nitter = 1;
TxGainRF = 0:6:60;

%% GENERATE BIT INFORMASI
txdata_s1 = randi(MOD_ORDER, 1, N_DATA_SYMS) - 1;
txdata_s2 = randi(MOD_ORDER, 1, N_DATA_SYMS) - 1;

for k = TxGainRF
    error_nc1 = [] ;ber_nc1 = [];
    error_nc2 = [] ;ber_nc2 = [];
    Tnc = [];
    fprintf('NC - Menghitung BER @ SNR = %d\n',k);
    for itter = 1:Nitter;
        tic;
        %% TIMESLOT 1
        rxdata_relay1 = tx_ofdm_siso(txdata_s1,k,1,2);
        rxdata_relay1 = rx_ofdm(rxdata_relay1);
        RSSI_NC1 = [RSSI_NC1 rx_RSSI];

        %% TIMESLOT 2
        rxdata_relay2 = tx_ofdm_siso(txdata_s2,k,3,2);
        rxdata_relay2 = rx_ofdm(rxdata_relay2);
        RSSI_NC2 = [RSSI_NC2 rx_RSSI];

        %% TIMESLOT 3
        rxdata_relay1_biner =
de2bi(rxdata_relay1,'left-msb');
        rxdata_relay2_biner =
de2bi(rxdata_relay2,'left-msb');
        data_xor =
xor(rxdata_relay1_biner,rxdata_relay2_biner);
        txdata_relay = bi2de(data_xor,'left-msb').';

        [rxdata_s1,rxdata_s2] =
tx_ofdm_simo(txdata_relay,k,2,1,3);
        rxdata_s1 = rx_ofdm(rxdata_s1);
```

```

        rxdata_s2 = rx_ofdm(rxdata_s2);

        rxdata_s1 = de2bi(rxdata_s1, 'left-msb');
        rxdata_s1 =
xor(rxdata_s1, rxdata_relay1_biner);
        rxdata_s1 = bi2de(rxdata_s1, 'left-msb').';

        rxdata_s2 = de2bi(rxdata_s2, 'left-msb');
        rxdata_s2 =
xor(rxdata_s2, rxdata_relay2_biner);
        rxdata_s2 = bi2de(rxdata_s2, 'left-msb').';
        Tnc = [Tnc toc];
        %% RESULT
        sym_errs = sum(txdata_s1 ~= rxdata_s2);
        bit_errs =
length(find(dec2bin(bitxor(txdata_s1, rxdata_s2), 8) ==
'1'));

        error_nc1 = [error_nc1
sum(xor(de2bi(rxdata_relay1, 'left-
msb'), de2bi(rxdata_s2, 'left-msb')))];
        error_nc2 = [error_nc2
sum(xor(de2bi(rxdata_relay2, 'left-
msb'), de2bi(rxdata_s1, 'left-msb')))];
        ber_nc1 = [ber_nc1
length(find(dec2bin(bitxor(txdata_s1, rxdata_s2), 8) ==
'1'))/(length(txdata_s1)*log2(MOD_ORDER))];
        ber_nc2 = [ber_nc2
length(find(dec2bin(bitxor(txdata_s2, rxdata_s1), 8) ==
'1'))/(length(txdata_s1)*log2(MOD_ORDER))];
        end
        T_NC = [T_NC mean(Tnc)];
        TR_NC = [TR_NC (length(txdata_s1)-
bit_errs)/mean(Tnc)];
        BER_NC1 = [BER_NC1 mean(ber_nc1)];
        BER_NC2 = [BER_NC2 mean(ber_nc2)];
end
for k = TxGainRF
error_tc1 = [] ;ber_tc1 = [];
error_tc2 = [] ;ber_tc2 = [];
Ttc= [];
    fprintf('non NC - Menghitung BER @ SNR = %d\n', k);
    for itter = 1:Nitter;
        tic;
        %% TIMESLOT 1
        rxdata_relay1 = tx_ofdm_siso(txdata_s1, k, 1, 2);
        rxdata_relay1 = rx_ofdm(rxdata_relay1);
        RSSI_TC1 = [RSSI_TC1 rx_RSSI];
    end
end

```

```

%% TIMESLOT 2
rxdata_s2 = tx_ofdm_siso(rxdata_relay1,k,2,3);
rxdata_s2 = rx_ofdm(rxdata_s2);

%% TIMESLOT 3
rxdata_relay2 = tx_ofdm_siso(txdata_s2,k,3,2);
rxdata_relay2 = rx_ofdm(rxdata_relay2);
RSSI_TC2 = [RSSI_TC2 rx_RSSI];

%% TIMESLOT 4
rxdata_s1 = tx_ofdm_siso(rxdata_relay2,k,2,1);
rxdata_s1 = rx_ofdm(rxdata_s1);
Ttc = [Ttc toc];

%% RESULT
sym_errs = sum(txdata_s1 ~= rxdata_s2);
bit_errs =
length(find(dec2bin(bitxor(txdata_s1, rxdata_s2),8) ==
'1'));

    error_tc1 = [error_tc1
sum(xor(de2bi(rxdata_relay1,'left-
msb'),de2bi(rxdata_s2,'left-msb')))]];
    error_tc2 = [error_tc1
sum(xor(de2bi(rxdata_relay2,'left-
msb'),de2bi(rxdata_s1,'left-msb')))]];
    ber_tc1 = [ber_tc1
length(find(dec2bin(bitxor(txdata_s1, rxdata_s2),8) ==
'1'))/(length(txdata_s1)*log2(MOD_ORDER))];
    ber_tc2 = [ber_tc2
length(find(dec2bin(bitxor(txdata_s2, rxdata_s1),8) ==
'1'))/(length(txdata_s1)*log2(MOD_ORDER))];
    end
    T_TC = [T_TC mean(Ttc)];
    TR_TC = [TR_TC (length(txdata_s1)-
bit_errs)/mean(Ttc)];
    BER_TC1 = [BER_TC1 mean(ber_tc1)];
    BER_TC2 = [BER_TC2 mean(ber_tc2)];
end
if(USE_WARPLAB_TXRX==0)
    TxGainRF = TxGainRF-3;
end
for k = TxGainRF
error_dr1 = [] ;ber_dr1 = [];
error_dr2 = [] ;ber_dr2 = [];
Tdr=[];
    fprintf('Direct - Menghitung BER @ SNR = %d\n',k);
    for itter = 1:Nitter;
        tic;

```

```

%% TIMESLOT 1
rxdata_s2 = tx_ofdm_siso(txdata_s1,k,1,3);
rxdata_s2 = rx_ofdm(rxdata_s2);
RSSI_DR1 = [RSSI_DR1 rx_RSSI];

%% TIMESLOT 2
rxdata_s1 = tx_ofdm_siso(txdata_s2,k,3,1);
rxdata_s1 = rx_ofdm(rxdata_s1);
RSSI_DR2 = [RSSI_DR2 rx_RSSI];
Tdr=[Tdr toc];

%% RESULT
sym_errs = sum(txdata_s1 ~= rxdata_s2);
bit_errs =
length(find(dec2bin(bitxor(txdata_s1, rxdata_s2),8) ==
'1')));

error_dr1 = [error_dr1
sum(xor(de2bi(rxdata_relay1,'left-
msb'),de2bi(rxdata_s2,'left-msb')))]);
error_dr2 = [error_dr1
sum(xor(de2bi(rxdata_relay2,'left-
msb'),de2bi(rxdata_s1,'left-msb')))]);
ber_dr1 = [ber_dr1
length(find(dec2bin(bitxor(txdata_s1, rxdata_s2),8) ==
'1'))/(length(txdata_s1)*log2(MOD_ORDER))];
ber_dr2 = [ber_dr2
length(find(dec2bin(bitxor(txdata_s2, rxdata_s1),8) ==
'1'))/(length(txdata_s1)*log2(MOD_ORDER))];
end
T_DR = [T_DR mean(Tdr)];
TR_DR = [TR_DR (length(txdata_s1)-
bit_errs)/mean(Tdr)];
BER_DR1 = [BER_DR1 mean(ber_dr1)];
BER_DR2 = [BER_DR2 mean(ber_dr2)];
end
tx_gain_set = [8;16;24;32;40;48;56;63];
eqv_gain = [-31;-28;-23;-19;-14.5;-11.5;-7;-2.5];
tx_outp =
interp1(tx_gain_set,eqv_gain,TxGainRF,'linear','extrap
');

figure(1)
semilogy(tx_outp,BER_NC1,'bo-',tx_outp,BER_TC1,'ro-
',tx_outp,BER_DR1,'ko-'); grid on;
title('Performa BER');
xlabel('Power Transmit (dBm)');
ylabel('BER');

```



```

legend('Network coding', 'Multihop
konvensional', 'Direct S1 ke S2');
figure(2)
plot(tx_outp, T_NC, 'bo-', tx_outp, T_TC, 'ro-'); grid on;
title('Waktu satu cycle TX/RX');
xlabel('Power Transmit (dBm)');
ylabel('Second');
legend('Network coding', 'Multihop konvensional');
figure(3)
plot(tx_outp, TR_NC/1e3, 'bo-', tx_outp, TR_TC/1e3, 'ro-');
grid on;
title('Throughput');
xlabel('Power Transmit (dBm)');
ylabel('Kbps');
legend('Network coding', 'Multihop konvensional');

%% 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\BAMBANG EKO
SURYA\HASIL';
if(USE_WARPLAB_TXRX)
loc_fix = strcat
(loc, '\', 'Implementasi_', hari, bulan, tahun, '_', jam, '.',
menit, '\');
else
loc_fix = strcat
(loc, '\', 'Simulasi_', hari, bulan, tahun, '_', jam, '.', meni
t, '\');
end
mkdir (loc_fix);
save([loc_fix nama_file1 '.mat']);

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

```

```

nama_file3 = 'Grafik_Throughput';
saveas(figure(2),[loc_fix nama_file3 '.fig']);
print(figure(2), '-dbitmap',[loc_fix nama_file3
'.bmp']);

nama_file4 = 'Grafik_Durasi';
saveas(figure(3),[loc_fix nama_file4 '.fig']);
print(figure(3), '-dbitmap',[loc_fix nama_file4
'.bmp']);
else
    return;
end

```

rx_ofdm.m

```

function rx_data = rx_ofdm(rx_vec_air)

global N_OFDM_SYMS MOD_ORDER SC_IND_PILOTS SC_IND_DATA
N_SC CP_LEN ...
    N_DATA_SYMS FFT_OFFSET LTS_CORR_THRESH
DO_APPLY_CFO_CORRECTION ...
    DO_APPLY_PHASE_ERR_CORRECTION
DO_APPLY_SFO_CORRECTION DECIMATE_RATE

% Define a half-band 2x interpolation filter response
interp_filt2 = zeros(1,43);
interp_filt2([1 3 5 7 9 11 13 15 17 19 21]) = [12 -32
72 -140 252 -422 682 -1086 1778 -3284 10364];
interp_filt2([23 25 27 29 31 33 35 37 39 41 43]) =
interp_filt2(fliplr([1 3 5 7 9 11 13 15 17 19 21]));
interp_filt2(22) = 16384;
interp_filt2 = interp_filt2./max(abs(interp_filt2));

% Define the preamble
sts_f = zeros(1,64);
sts_f(1:27) = [0 0 0 0 -1-1i 0 0 0 -1-1i 0 0 0 1+1i 0
0 0 1+1i 0 0 0 1+1i 0 0 0 1+1i 0 0];
sts_f(39:64) = [0 0 1+1i 0 0 0 -1-1i 0 0 0 1+1i 0 0 0
-1-1i 0 0 0 -1-1i 0 0 0 1+1i 0 0 0];
sts_t = ifft(sqrt(13/6).*sts_f, 64);
sts_t = sts_t(1:16);

% LTS for CFO and channel estimation
lts_f = [0 1 -1 -1 1 1 -1 1 -1 1 -1 -1 -1 -1 1 1 -1
-1 1 -1 1 -1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 -1 -1 1
1 -1 1 -1 1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 1 1 1 1];
lts_t = ifft(lts_f, 64);

```

```

% Define the pilot tones
pilots = [1 1 -1 1].';

% Repeat the pilots across all OFDM symbols
pilots_mat = repmat(pilots, 1, N_OFDM_SYMS);
%% Decimate
if(DECIMATE_RATE == 1)
    raw_rx_dec = rx_vec_air;
elseif(DECIMATE_RATE == 2)
    raw_rx_dec = filter(interp_filt2, 1, rx_vec_air);
    raw_rx_dec = raw_rx_dec(1:2:end);
end

%% Correlate for LTS

% Complex cross correlation of Rx waveform with time-
domain LTS
lts_corr = abs(conv(conj(fliplr(lts_t)),
    sign(raw_rx_dec)));

% Skip early and late samples
lts_corr = lts_corr(32:end-32);

% Find all correlation peaks
lts_peaks = find(lts_corr >
    LTS_CORR_THRESH*max(lts_corr));

% Select best candidate correlation peak as LTS-
payload boundary
[LTS1, LTS2] = meshgrid(lts_peaks,lts_peaks);
[lts_second_peak_index,y] = find(LTS2-LTS1 ==
    length(lts_t));

% Stop if no valid correlation peak was found
if isempty(lts_second_peak_index)
    fprintf('No LTS Correlation Peaks Found!\n');
    lts_peaks = [566 630];
    lts_second_peak_index = numel(lts_peaks);
    % return;
end

% Set the sample indices of the payload symbols and
preamble
payload_ind =
    lts_peaks(max(lts_second_peak_index))+32;
lts_ind = payload_ind-160;

if(DO_APPLY_CFO_CORRECTION)
    %Extract LTS (not yet CFO corrected)

```

```

rx_lts = raw_rx_dec(lts_ind : lts_ind+159);
rx_lts1 = rx_lts(-64+FFT_OFFSET + [97:160]);
rx_lts2 = rx_lts(-FFT_OFFSET + [97:160]);

%Calculate coarse CFO est
rx_cfo_est_lts = mean(unwrap(angle(rx_lts2 .*
conj(rx_lts1))));
rx_cfo_est_lts = rx_cfo_est_lts/(2*pi*64);
else
rx_cfo_est_lts = 0;
end

% Apply CFO correction to raw Rx waveform
rx_cfo_corr_t = exp(-
1i*2*pi*rx_cfo_est_lts*[0:length(raw_rx_dec)-1]);
rx_dec_cfo_corr = raw_rx_dec .* rx_cfo_corr_t;

% Re-extract LTS for channel estimate
rx_lts = rx_dec_cfo_corr(lts_ind : lts_ind+159);
rx_lts1 = rx_lts(-64+FFT_OFFSET + [97:160]);
rx_lts2 = rx_lts(-FFT_OFFSET + [97:160]);

rx_lts1_f = fft(rx_lts1);
rx_lts2_f = fft(rx_lts2);

% Calculate channel estimate
rx_H_est = lts_f .* (rx_lts1_f + rx_lts2_f)/2;

= [rx_dec_cfo_corr zeros(1,8000)];
payload_vec = rx_dec_cfo_corr(payload_ind :
payload_ind+N_OFDM_SYMS*(N_SC+CP_LEN)-1);
payload_mat = reshape(payload_vec, (N_SC+CP_LEN),
N_OFDM_SYMS);

% Remove the cyclic prefix, keeping FFT_OFFSET samples
of CP (on average)
payload_mat_noCP = payload_mat(CP_LEN-
FFT_OFFSET+[1:N_SC], :);

% Take the FFT
syms_f_mat = fft(payload_mat_noCP, N_SC, 1);

% Equalize (zero-forcing, just divide by compled chan
estimates)
syms_eq_mat = syms_f_mat ./ repmat(rx_H_est.', 1,
N_OFDM_SYMS);

if DO_APPLY_SFO_CORRECTION
pilots_f_mat = syms_eq_mat(SC_IND_PILOTS, :);

```

```

    pilots_f_mat_comp = pilots_f_mat.*pilots_mat;
    pilot_phases =
unwrap(angle(fftshift(pilots_f_mat_comp,1)), [],1);
    b = mean(diff(pilot_phases) ./
repmat(mod(diff(fftshift(SC_IND_PILOTS)),64) .',1,N_OFDM
M_SYMS));
    pilot_phase_sfo_corr = fftshift((-32:31) .*b,1);
    pilot_phase_corr = exp(-
1i*(pilot_phase_sfo_corr));

    % Apply the pilot phase correction per symbol
    syms_eq_mat = syms_eq_mat .* pilot_phase_corr;
else
    pilot_phase_sfo_corr = zeros(N_SC, N_OFDM_SYMS);
end

if DO_APPLY_PHASE_ERR_CORRECTION
    % Extract the pilots and calculate per-symbol
phase error
    pilots_f_mat = syms_eq_mat(SC_IND_PILOTS, :);
    pilots_f_mat_comp = pilots_f_mat.*pilots_mat;
    pilot_phase_err = angle(mean(pilots_f_mat_comp));
else
    pilot_phase_err = zeros(1,N_OFDM_SYMS);
end
pilot_phase_err_corr = repmat(pilot_phase_err, N_SC,
1);
pilot_phase_corr = exp(-1i*(pilot_phase_err_corr));

% Apply the pilot phase correction per symbol
syms_eq_pc_mat = syms_eq_mat .* pilot_phase_corr;
payload_syms_mat = syms_eq_pc_mat(SC_IND_DATA, :);

%% Demodulate
rx_syms = reshape(payload_syms_mat, 1, N_DATA_SYMS);

demod_fcn_bpsk = @(x) double(real(x)>0);
demod_fcn_qpsk = @(x) double(2*(real(x)>0) +
1*(imag(x)>0));
demod_fcn_16qam = @(x) (8*(real(x)>0) +
(4*(abs(real(x))<0.6325)) + (2*(imag(x)>0)) +
(1*(abs(imag(x))<0.6325)));
demod_fcn_64qam = @(x) (32*(real(x)>0) +
(16*(abs(real(x))<0.6172)) +
(8*((abs(real(x))<0.9258) &&((abs(real(x))>(0.3086))
)) + (4*(imag(x)>0)) + (2*(abs(imag(x))<0.6172)) +
(1*((abs(imag(x))<0.9258) &&((abs(imag(x))>(0.3086))
))););

```

```

switch(MOD_ORDER)
    case 2           % BPSK
        rx_data = arrayfun(demod_fcn_bpsk, rx_syms);
    case 4           % QPSK
        rx_data = arrayfun(demod_fcn_qpsk, rx_syms);
    case 16          % 16-QAM
        rx_data = arrayfun(demod_fcn_16qam, rx_syms);
    case 64          % 64-QAM
        rx_data = arrayfun(demod_fcn_64qam, rx_syms);
end

```

tx ofdm siso

```

function [rx_vec_air] =
tx_ofdm_siso(tx_data, SNR, NODE_TX, NODE_RX)

global USE_WARPLAB_TXRX CHANNEL N_OFDM_SYMS MOD_ORDER
...
    TX_SCALE INTERP_RATE SC_IND_PILOTS SC_IND_DATA
N_SC CP_LEN ...
    USE_AGC MAX_TX_LEN SAMP_PADDING rx_RSSI

if(USE_WARPLAB_TXRX)

    USE_AGC = 0;

    NUMNODES = 3;

    %Create a vector of node objects
    nodes = wl_initNodes(3);

    %Create a UDP broadcast trigger and tell each node
to be ready for it
    eth_trig = wl_trigger_eth_udp_broadcast;

    wl_triggerManagerCmd(nodes, 'add_ethernet_trigger', [eth
_trig]);

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

    %Set up the interface for the experiment
    wl_interfaceCmd(nodes, 'RF_ALL', 'tx_gains', 3, SNR);

```

```

wl_interfaceCmd(nodes, 'RF_ALL', 'channel', 2.4, CHANNEL);
    if (USE_AGC)

wl_interfaceCmd(nodes, 'RF_ALL', 'rx_gain_mode', 'automatic');
        wl_basebandCmd(nodes, 'agc_target', -10);
        wl_basebandCmd(nodes, 'agc_trig_delay', 511);
    else

wl_interfaceCmd(nodes, 'RF_ALL', 'rx_gain_mode', 'manual');
        RxGainRF = 1; %Rx RF Gain in [1:3]
        RxGainBB = 3; %Rx Baseband Gain in [0:31]

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

    maximum_buffer_len =
nodes(NODE_TX).baseband.txIQLen;
    SAMP_FREQ =
wl_basebandCmd(nodes(NODE_TX), 'tx_buff_clk_freq');
    node_tx = nodes(NODE_TX);
    node_rx = nodes(NODE_RX);
    RF_TX = RFA;
    RF_RX = RFA;
txLength = node_tx.baseband.txIQLen;
Ts_RSSI =
1/(wl_basebandCmd(node_rx, 'rx_rssi_clk_freq'));
Ts = 1/(wl_basebandCmd(node_tx, 'tx_buff_clk_freq'));

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

wl_basebandCmd(nodes, 'tx_length', maximum_buffer_len);
    example_mode_string = 'hw';
else
    % Use sane defaults for hardware-dependent params
in sim-only version
    maximum_buffer_len = min(MAX_TX_LEN, 2^20);
    SAMP_FREQ = 40e6;
    example_mode_string = 'sim';
end

%% Define a half-band 2x interpolation filter response
interp_filt2 = zeros(1, 43);

```

```

interp_filt2([1 3 5 7 9 11 13 15 17 19 21]) = [12 -32
72 -140 252 -422 682 -1086 1778 -3284 10364];
interp_filt2([23 25 27 29 31 33 35 37 39 41 43]) =
interp_filt2(fliplr([1 3 5 7 9 11 13 15 17 19 21]));
interp_filt2(22) = 16384;
interp_filt2 = interp_filt2./max(abs(interp_filt2));

% Define the preamble
sts_f = zeros(1,64);
sts_f(1:27) = [0 0 0 0 -1-1i 0 0 0 -1-1i 0 0 0 1+1i 0
0 0 1+1i 0 0 0 1+1i 0 0 0 1+1i 0 0];
sts_f(39:64) = [0 0 1+1i 0 0 0 -1-1i 0 0 0 1+1i 0 0 0
-1-1i 0 0 0 -1-1i 0 0 0 1+1i 0 0 0];
sts_t = ifft(sqrt(13/6).*sts_f, 64);
sts_t = sts_t(1:16);

% LTS for CFO and channel estimation
lts_f = [0 1 -1 -1 1 1 -1 1 -1 1 -1 -1 -1 -1 1 1 -1
-1 1 -1 1 -1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 -1 -1 1
1 -1 1 -1 1 1 1 1 1 -1 -1 1 1 -1 1 -1 1 1 1 1];
lts_t = ifft(lts_f, 64);

% Use 30 copies of the 16-sample STS for extra AGC
settling margin
preamble = [repmat(sts_t, 1, 30) lts_t(33:64) lts_t
lts_t];

% Sanity check inputs
if(SAMP_PADDING + INTERP_RATE*((N_OFDM_SYMS * (N_SC +
CP_LEN)) + length(preamble) + 100) >
maximum_buffer_len)
    fprintf('Too many OFDM symbols for
TX_NUM_SAMPS!\n');
    fprintf('Raise TX_NUM_SAMPS to %d, or \n',
SAMP_PADDING + INTERP_RATE*((N_OFDM_SYMS * (N_SC +
CP_LEN)) + length(preamble) + 100));
    fprintf('Reduce N_OFDM_SYMS to %d\n', floor((
(maximum_buffer_len/INTERP_RATE)-length(preamble) )/(
N_SC + CP_LEN )) - 1);
    return;
end

%% Generate a payload
tx_data = tx_data;

% Functions for data -> complex symbol mapping (avoids
comm toolbox requirement for qammod)
modvec_bpsk = (1/sqrt(2)) .* [-1 1];
modvec_16qam = (1/sqrt(10)) .* [-3 -1 +3 +1];

```



```

modvec_64qam = (1/sqrt(43)) .* [-7 -5 -1 -3 +7 +5 +1
+3];

mod_fcn_bpsk = @(x) complex(modvec_bpsk(1+x),0);
mod_fcn_qpsk = @(x) complex(modvec_bpsk(1+bitshift(x,
-1)), modvec_bpsk(1+mod(x, 2)));
mod_fcn_16qam = @(x)
complex(modvec_16qam(1+bitshift(x, -2)),
modvec_16qam(1+mod(x,4)));
mod_fcn_64qam = @(x)
complex(modvec_64qam(1+bitshift(x, -3)),
modvec_64qam(1+mod(x,8)));

% Map the data values on to complex symbols
switch MOD_ORDER
    case 2 % BPSK
        tx_syms = arrayfun(mod_fcn_bpsk, tx_data);
    case 4 % QPSK
        tx_syms = arrayfun(mod_fcn_qpsk, tx_data);
    case 16 % 16-QAM
        tx_syms = arrayfun(mod_fcn_16qam, tx_data);
    case 64 % 64-QAM
        tx_syms = arrayfun(mod_fcn_64qam, tx_data);
    otherwise
        fprintf('Invalid MOD_ORDER (%d)! Must be in
[2, 4, 16]\n', MOD_ORDER);
        return;
end

% Reshape the symbol vector to a matrix with one
column per OFDM symbol
tx_syms_mat = reshape(tx_syms, length(SC_IND_DATA),
N_OFDM_SYMS);

% Define the pilot tones
pilots = [1 1 -1 1].';

% Repeat the pilots across all OFDM symbols
pilots_mat = repmat(pilots, 1, N_OFDM_SYMS);

%% IFFT

% Construct the IFFT input matrix
ifft_in_mat = zeros(N_SC, N_OFDM_SYMS);

% Insert the data and pilot values; other subcarriers
will remain at 0
ifft_in_mat(SC_IND_DATA, :) = tx_syms_mat;
ifft_in_mat(SC_IND_PILOTS, :) = pilots_mat;

```

```

%Perform the IFFT
tx_payload_mat = ifft(iff_t_in_mat, N_SC, 1);

% Insert the cyclic prefix
if(CP_LEN > 0)
    tx_cp = tx_payload_mat((end-CP_LEN+1 : end), :);
    tx_payload_mat = [tx_cp; tx_payload_mat];
end

% Reshape to a vector
tx_payload_vec = reshape(tx_payload_mat, 1,
    numel(tx_payload_mat));

% Construct the full time-domain OFDM waveform
tx_vec = [preamble tx_payload_vec];

% Pad with zeros for transmission
tx_vec_padded = [tx_vec, zeros(1,50)];

%% Interpolate
if(INTERP_RATE == 1)
    tx_vec_air = tx_vec_padded;
elseif(INTERP_RATE == 2)
    tx_vec_2x = zeros(1, 2*numel(tx_vec_padded));
    tx_vec_2x(1:2:end) = tx_vec_padded;
    tx_vec_air = filter(interp_filt2, 1, tx_vec_2x);
end

% Scale the Tx vector
tx_vec_air = TX_SCALE .* tx_vec_air ./
    max(abs(tx_vec_air));

TX_NUM_SAMPS = length(tx_vec_air);

if(USE_WARPLAB_TXRX)
    wl_basebandCmd(nodes, 'tx_delay', 0);
    wl_basebandCmd(nodes, 'tx_length',
TX_NUM_SAMPS+100); % Number of
samples to send
    wl_basebandCmd(nodes, 'rx_length',
TX_NUM_SAMPS+SAMP_PADDING); % Number of samples
to receive
end

%% WARPLab Tx/Rx
if(USE_WARPLAB_TXRX)
    % Write the Tx waveform to the Tx node

```

```

    wl_basebandCmd(node_tx, RF_TX, 'write_IQ',
tx_vec_air(:));

    % Enable the Tx and Rx radios
    wl_interfaceCmd(node_tx, RF_TX, 'tx_en');
    wl_interfaceCmd(node_rx, RF_RX, 'rx_en');

    % Enable the Tx and Rx buffers
    wl_basebandCmd(node_tx, RF_TX, 'tx_buff_en');
    wl_basebandCmd(node_rx, RF_RX, 'rx_buff_en');

    % Trigger the Tx/Rx cycle at both nodes
    eth_trig.send();

    rx_RSSI =
wl_basebandCmd(node_rx, [RF_RX], 'read_RSSI', 0, txLength/
(Ts_RSSI/Ts));
    % Retrieve the received waveform from the Rx node
    rx_vec_air = wl_basebandCmd(node_rx, RF_RX,
'read_IQ');

    rx_vec_air = rx_vec_air(:).';

    %Disable the Tx/Rx radios and buffers
    wl_basebandCmd(nodes, 'RF_ALL', 'tx_rx_buff_dis');
    wl_interfaceCmd(nodes, 'RF_ALL', 'tx_rx_dis');
else
    % Mode Simulasi
    % Perfect (ie. Rx=Tx):
    % rx_vec_air = tx_vec_air;
    % AWGN:
    rx_vec_air = awgn(tx_vec_air, SNR, 'measured');
End

```

tx_ofdm_simo

```

function [rx_vec_air1,rx_vec_air2] =
tx_ofdm_simo(tx_data,SNR,NODE_TX,NODE_RX1,NODE_RX2)

global USE_WARPLAB_TXRX CHANNEL N_OFDM_SYMS MOD_ORDER
...
    TX_SCALE INTERP_RATE SC_IND_PILOTS SC_IND_DATA
N_SC CP_LEN ...
    USE_AGC MAX_TX_LEN SAMP_PADDING

if(USE_WARPLAB_TXRX)

    USE_AGC = 0;

```

```

NUMNODES = 3;

%Create a vector of node objects
nodes = wl_initNodes(3);

%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(NODE_TX));
[RFA1,RFB] = wl_getInterfaceIDs(nodes(NODE_RX1));
[RFA2,RFB] = wl_getInterfaceIDs(nodes(NODE_RX2));

%Set up the interface for the experiment
wl_interfaceCmd(nodes, 'RF_ALL', 'tx_gains', 3, SNR);

wl_interfaceCmd(nodes, 'RF_ALL', 'channel', 2.4, CHANNEL);

if(USE_AGC)

wl_interfaceCmd(nodes, 'RF_ALL', 'rx_gain_mode', 'automat
ic');
    wl_basebandCmd(nodes, 'agc_target', -10);
    wl_basebandCmd(nodes, 'agc_trig_delay', 511);
else

wl_interfaceCmd(nodes, 'RF_ALL', 'rx_gain_mode', 'manual'
);
    RxGainRF = 1; %Rx RF Gain in [1:3]
    RxGainBB = 3; %Rx Baseband Gain in [0:31]

wl_interfaceCmd(nodes, 'RF_ALL', 'rx_gains', RxGainRF, RxG
ainBB);
end

maximum_buffer_len =
nodes(NODE_TX).baseband.txIQLen;
SAMP_FREQ =
wl_basebandCmd(nodes(NODE_TX), 'tx_buff_clk_freq');
node_tx = nodes(NODE_TX);
node_rx1 = nodes(NODE_RX1);
node_rx2 = nodes(NODE_RX2);
RF_TX = RFA;
RF_RX1 = RFA1;
RF_RX2 = RFA2;

```

```

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

wl_basebandCmd(nodes, 'tx_length', maximum_buffer_len);
    example_mode_string = 'hw';
else
    % Use sane defaults for hardware-dependent params
in sim-only version
    maximum_buffer_len = min(MAX_TX_LEN, 2^20);
    SAMP_FREQ           = 40e6;
    example_mode_string = 'sim';
end

%% Define a half-band 2x interpolation filter response
interp_filt2 = zeros(1,43);
interp_filt2([1 3 5 7 9 11 13 15 17 19 21]) = [12 -32
72 -140 252 -422 682 -1086 1778 -3284 10364];
interp_filt2([23 25 27 29 31 33 35 37 39 41 43]) =
interp_filt2(fliplr([1 3 5 7 9 11 13 15 17 19 21]));
interp_filt2(22) = 16384;
interp_filt2 = interp_filt2./max(abs(interp_filt2));

% Define the preamble
sts_f = zeros(1,64);
sts_f(1:27) = [0 0 0 0 -1-1i 0 0 0 -1-1i 0 0 0 1+1i 0
0 0 1+1i 0 0 0 1+1i 0 0 0 1+1i 0 0];
sts_f(39:64) = [0 0 1+1i 0 0 0 -1-1i 0 0 0 1+1i 0 0 0
-1-1i 0 0 0 -1-1i 0 0 0 1+1i 0 0 0];
sts_t = ifft(sqrt(13/6).*sts_f, 64);
sts_t = sts_t(1:16);

% LTS for CFO and channel estimation
lts_f = [0 1 -1 -1 1 1 -1 1 -1 1 -1 -1 -1 -1 1 1 -1
-1 1 -1 1 -1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 -1 -1 1
1 -1 1 -1 1 1 1 1 1 1 -1 -1 1 1 -1 1 1 1];
lts_t = ifft(lts_f, 64);

% Use 30 copies of the 16-sample STS for extra AGC
settling margin
preamble = [repmat(sts_t, 1, 30) lts_t(33:64) lts_t
lts_t];

% Sanity check inputs
if(SAMP_PADDING + INTERP_RATE*(N_OFDM_SYMS * (N_SC +
CP_LEN)) + length(preamble) + 100) >
maximum_buffer_len)
    fprintf('Too many OFDM symbols for
TX_NUM_SAMPS!\n');

```

```

    fprintf('Raise TX_NUM_SAMPS to %d, or \n',
    SAMP_PADDING + INTERP_RATE*(N_OFDM_SYMS * (N_SC +
    CP_LEN)) + length(preamble) + 100));
    fprintf('Reduce N_OFDM_SYMS to %d\n', floor((
    (maximum_buffer_len/INTERP_RATE)-length(preamble) )/(
    N_SC + CP_LEN )) - 1);
    return;
end

%% Generate a payload
tx_data = tx_data;

% Functions for data -> complex symbol mapping (avoids
comm toolbox requirement for qammod)
modvec_bpsk = (1/sqrt(2)) .* [-1 1];
modvec_16qam = (1/sqrt(10)) .* [-3 -1 +3 +1];
modvec_64qam = (1/sqrt(43)) .* [-7 -5 -1 -3 +7 +5 +1
+3];

mod_fcn_bpsk = @(x) complex(modvec_bpsk(1+x),0);
mod_fcn_qpsk = @(x) complex(modvec_bpsk(1+bitshift(x,
-1)), modvec_bpsk(1+mod(x, 2)));
mod_fcn_16qam = @(x)
complex(modvec_16qam(1+bitshift(x, -2)),
modvec_16qam(1+mod(x,4)));
mod_fcn_64qam = @(x)
complex(modvec_64qam(1+bitshift(x, -3)),
modvec_64qam(1+mod(x,8)));

% Map the data values on to complex symbols
switch MOD_ORDER
    case 2 % BPSK
        tx_syms = arrayfun(mod_fcn_bpsk, tx_data);
    case 4 % QPSK
        tx_syms = arrayfun(mod_fcn_qpsk, tx_data);
    case 16 % 16-QAM
        tx_syms = arrayfun(mod_fcn_16qam, tx_data);
    case 64 % 64-QAM
        tx_syms = arrayfun(mod_fcn_64qam, tx_data);
    otherwise
        fprintf('Invalid MOD_ORDER (%d)! Must be in
[2, 4, 16]\n', MOD_ORDER);
        return;
end

% Reshape the symbol vector to a matrix with one
column per OFDM symbol
tx_syms_mat = reshape(tx_syms, length(SC_IND_DATA),
N_OFDM_SYMS);

```

```

% Define the pilot tones
pilots = [1 1 -1 1].';

% Repeat the pilots across all OFDM symbols
pilots_mat = repmat(pilots, 1, N_OFDM_SYMS);

%% IFFT

% Construct the IFFT input matrix
ifft_in_mat = zeros(N_SC, N_OFDM_SYMS);

% Insert the data and pilot values; other subcarriers
will remain at 0
ifft_in_mat(SC_IND_DATA, :) = tx_syms_mat;
ifft_in_mat(SC_IND_PILOTS, :) = pilots_mat;

%Perform the IFFT
tx_payload_mat = ifft(ifft_in_mat, N_SC, 1);

% Insert the cyclic prefix
if(CP_LEN > 0)
    tx_cp = tx_payload_mat((end-CP_LEN+1 : end), :);
    tx_payload_mat = [tx_cp; tx_payload_mat];
end

% Reshape to a vector
tx_payload_vec = reshape(tx_payload_mat, 1,
    numel(tx_payload_mat));

% Construct the full time-domain OFDM waveform
tx_vec = [preamble tx_payload_vec];

% Pad with zeros for transmission
tx_vec_padded = [tx_vec, zeros(1,50)];

%% Interpolate
if(INTERP_RATE == 1)
    tx_vec_air = tx_vec_padded;
elseif(INTERP_RATE == 2)
    tx_vec_2x = zeros(1, 2*numel(tx_vec_padded));
    tx_vec_2x(1:2:end) = tx_vec_padded;
    tx_vec_air = filter(interp_filt2, 1, tx_vec_2x);
end

% Scale the Tx vector
tx_vec_air = TX_SCALE .* tx_vec_air ./
    max(abs(tx_vec_air));

```

```

TX_NUM_SAMPS = length(tx_vec_air);

if(USE_WARPLAB_TXRX)
    wl_basebandCmd(nodes, 'tx_delay', 0);
    wl_basebandCmd(nodes, 'tx_length',
TX_NUM_SAMPS+100);          % Number of
samples to send
    wl_basebandCmd(nodes, 'rx_length',
TX_NUM_SAMPS+SAMP_PADDING); % Number of samples
to receive
end

%% WARPLab Tx/Rx
if(USE_WARPLAB_TXRX)
    % Write the Tx waveform to the Tx node
    wl_basebandCmd(node_tx, RF_TX, 'write_IQ',
tx_vec_air(:));

    % Enable the Tx and Rx radios
    wl_interfaceCmd(node_tx, RF_TX, 'tx_en');
    wl_interfaceCmd(node_rx1, RF_RX1, 'rx_en');
    wl_interfaceCmd(node_rx2, RF_RX2, 'rx_en');

    % Enable the Tx and Rx buffers
    wl_basebandCmd(node_tx, RF_TX, 'tx_buff_en');
    wl_basebandCmd(node_rx1, RF_RX1, 'rx_buff_en');
    wl_basebandCmd(node_rx2, RF_RX2, 'rx_buff_en');

    % Trigger the Tx/Rx cycle at both nodes
    eth_trig.send();

    % Retrieve the received waveform from the Rx node
    rx_vec_air1 = wl_basebandCmd(node_rx1, RF_RX1,
'read_IQ');
    rx_vec_air2 = wl_basebandCmd(node_rx2, RF_RX2,
'read_IQ');

    rx_vec_air1 = rx_vec_air1(:).';
    rx_vec_air2 = rx_vec_air2(:).';

    %Disable the Tx/Rx radios and buffers
    wl_basebandCmd(nodes, 'RF_ALL', 'tx_rx_buff_dis');
    wl_interfaceCmd(nodes, 'RF_ALL', 'tx_rx_dis');
else
    rx_vec_air1 = awgn(tx_vec_air, SNR, 'measured');
    rx_vec_air2 = awgn(tx_vec_air, SNR, 'measured');
end

```


Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] S. Weinstein and P. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Trans. Commun. Technol.*, vol. 19, no. 5, pp. 628–634, Oct. 1971.
- [2] X. Wang, Y. Xu, and Z. Feng, "Physical-layer network coding in OFDM system: Analysis and performance," in *2012 7th International ICST Conference on Communications and Networking in China (CHINACOM)*, 2012, pp. 139–143.
- [3] H. TRACEY and L. DESMOND S., *NETWORK CODING : An Introduction*. The Edinburgh Building, Cambridge CB2 8RU, UK: CAMBRIDGE UNIVERSITY PRESS, 2008.
- [4] Y. S. Cho, Jaekwon Kim, Won Young Yang, and Chung-Gu Kang, *MIMO-OFDM Wireless Communications with MATLAB*. Singapore: John Wiley & Sons (Asia) Pte Ltd, 2010.
- [5] M. H. Hayes, *Schaum's Outline of Theory and Problems of Digital Signal Processing*. USA, 1999.
- [6] J. G. Proakis and Masoud Salehi, *Digital Communications*. USA: The McGraw-Hill Companies, Inc, 2004.
- [7] "IEEE Xplore Abstract - Common phase error correction with feedback for OFDM in wireless communication." [Online]. Available:
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1188159&newsearch=true&queryText=Common%20Phase%20Error%20Correction%20with%20Feedback%20for%20OFDM>. [Accessed: 12-Dec-2015].
- [8] P. Murphy, Ashu Sabharwal, and Behnaam Aazhang, *Design of WARP: A Wireless Open-Access Research Platform*. Rice University: Department of Electrical and Computer Engineering.
- [9] M. Bhardwaj, Arun Gangwar, and Soni Devendra, "Review on OFDM: Concept, Scope & its Applications," *IOSR J. Mech. Civ. Eng. IOSRJMCE*, vol. 1, 2012.
- [10] T. Hwang, C. Yang, G. Wu, S. Li, and G. Y. Li, "OFDM and Its Wireless Applications: A Survey," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1673–1694, Mei 2009.
- [11] C. Fragouli, Soljanin, and Emina, "Network Coding Fundamentals, Foundation and Trends R in Networking," 2007.
- [12] C. L. Soung and Shengli Zhang, "Physical Layer Network Coding: Tutorial, Survey And Beyond," 2013.

- [13] "WARP Project." [Online]. Available: <http://warpproject.org/trac>. [Accessed: 15-Dec-2015].
- [14] Hasan, *IMPLEMENTASI DAN EVALUASI KINERJA ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING (OFDM) MENGGUNAKAN WARP*. Jurusan Teknik Elektro ITS, 2015.
- [15] "Fresnel zone," *Wikipedia, the free encyclopedia*. 03-Nov-2015.

DAFTAR RIWAYAT HIDUP



Bambang Eko Surya, lahir di Medan tanggal 11 Agustus 1992, merupakan anak pertama dari tiga bersaudara dari Bapak Irwan Tulus dan Ibu Ismiaty. Penulis menamatkan pendidikan dasar di SDN 015921 Kec. Kisaran Timur, Kabupaten Asahan. Kemudian melanjutkan pendidikan di SMP N 6 Kisaran, Kabupaten Asahan. Penulis memulai

pendidikannya di bidang telekomunikasi semenjak pendidikan menengah atasnya, penulis melanjutkan pendidikan di sekolah kejuruan SMK Telekomunikasi Sandhy Putra medan, jurusan Teknik Komputer dan Jaringan. Penulis kembali melanjutkan pendidikan tinggi tahun 2010 di Politeknik Negeri Bandung pada program studi D3 teknik Telekomunikasi dan lulus sebagai Ahli Madya tahun 2013. Pada tahun 2014 penulis melanjutkan pendidikan S1 melalui program Lintas Jalur ITS Jurusan Teknik Elektro Bidang Studi Telekomunikasi Multimedia. Pada tahun 2015 mengambil penelitian tugas akhir di laboratorium Telekomunikasi Multimedia B304 Jurusan Elektro, ITS.

Kontak :

Email : bambangekosurya@gmail.com