



**TUGAS AKHIR – TE 141599**

**IMPLEMENTASI DAN EVALUASI KINERJA  
ENCODER-DECODER HAMMING PADA *M-ARY*  
*QUADRATURE AMPLITUDE MODULATION* (M-QAM)  
MENGUNAKAN *WIRELESS OPEN-ACCESS RESEARCH*  
*PLATFORM* (WARP)**

**Rina Asmorowati  
NRP. 2213105024**

**Pembimbing  
Dr. Ir. Suwadi, MT.  
Dr. Ir. Titiek Suryani, MT.**

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



**FINAL PROJECT – TE 141599**

**IMPLEMENTATION AND PERFORMANCE  
EVALUATION OF HAMMING ENCODER-DECODER IN  
*M-ARY QUADRATURE AMPLITUDE MODULATION*  
(M-QAM) USING *WIRELESS OPEN-ACCESS RESEARCH  
PLATFORM (WARP)***

**Rina Asmorowati  
NRP. 2213105024**

**Advisors:  
Dr. Ir. Suwadi, MT.  
Dr. Ir. Titiek Suryani, MT.**

**DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty Of Industrial Technology  
Technology Institute of Sepuluh Nopember  
Surabaya 2015**


**IMPLEMENTASI DAN EVALUASI KINERJA  
ENCODER-DECODER HAMMING PADA  
M-ARY QUADRATURE AMPLITUDE MODULATION (M-QAM)  
MENGUNAKAN WIRELESS OPEN-ACCESS RESEARCH  
PLATFORM (WARP)**

**TUGAS AKHIR**

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

**Menyetujui :**

**Dosen Pembimbing I**



**Dr. Ir. Suwadi, MT.**  
NIP. 19680818 1993031 002

**Dosen Pembimbing II**



**Dr. Ir Titiek Suryani, MT.**  
NIP. 19641130 1989032 001



**Implementasi dan Evaluasi Kinerja Encoder-Decoder Hamming  
pada *M*-ary Quadrature Amplitude Modulation (M-QAM)  
Menggunakan *Wireless Open-Access Research Platform* (WARP)**

**Nama** : Rina Asmorowati  
**Pembimbing** : Dr. Ir. Suwadi, MT.  
Dr. Ir. Titiek Suryani, MT.

**ABSTRAK**

Salah satu kendala dalam sistem komunikasi digital adalah adanya *bit error* yang disebabkan oleh *noise* yang timbul selama proses transmisi berlangsung. Akibatnya data yang diterima pada penerima tidak sesuai dengan data yang dikirim. Apalagi jika modulasi yang digunakan adalah modulasi multilevel seperti *M*-ary Quadrature Amplitude Modulation (M-QAM). Semakin tinggi level modulasi maka tingkat kesalahannya pun semakin besar.

Kode Hamming merupakan salah satu *error control coding* yang berfungsi untuk mendeteksi kesalahan (*error detection*) sekaligus memperbaiki kesalahan (*error correction*) bit. Kode Hamming mampu untuk mengoreksi kesalahan tunggal dalam satu blok. Pada Tugas Akhir ini, kode Hamming (7,4), (15,11), dan (31,26) diimplementasikan pada M-QAM menggunakan modul *Wireless Open-Access Research Platform* (WARP) sebagai model pemancar dan penerima yang bersifat *real-time*.

Hasil implementasi dan pengukuran pada sistem modulasi M-QAM dengan menggunakan kode Hamming menunjukkan bahwa *Bit Error Rate* (BER) menurun. Kode Hamming dapat bekerja dengan baik pada level modulasi 4-QAM, 8-QAM dan 16-QAM. Tingginya level modulasi menyebabkan peluang *error* semakin besar, sehingga pada level modulasi 32-QAM dan 64-QAM, kinerja kode Hamming kurang maksimal karena kemampuan koreksinya terbatas hanya satu bit *error* pada setiap bloknya.

Kata kunci: Kode Hamming, M-QAM, WARP

*Halaman ini sengaja dikosongkan*

**Implementation and Performance Evaluation of Hamming  
Encoder-Decoder in *M*-ary Quadrature Amplitude Modulation  
(M-QAM) Using *Wireless Open-Access Research Platform (WARP)***

**Name** : Rina Asmorowati  
**Supervisors** : Dr. Ir. Suwadi, MT.  
Dr. Ir. Titiek Suryani, MT.

**ABSTRACT**

*One of the problems in digital communication systems is the presence of bit errors caused by noise that arises during the transmission process. As a result, the data received at the receiver is not in accordance with the data sent. Moreover, if the modulation used is multilevel modulation such as *M*-ary Quadrature Amplitude Modulation (M-QAM). Higher of modulation level is proportional with probability of errors.*

*Hamming code is one of error control coding is used to detect errors (error detection) while fixing errors (error correction) bits. Hamming code is able to correct a single error in one block. In this final project, the Hamming code (7,4), (15,11) and (31,26) in the M-QAM are implemented using *Wireless Open-Access Research Platform (WARP)* modules as a transmitter and receiver models that are real-time.*

*The implementation and results of measurements on M-QAM modulation system using Hamming code show that decrease Bit Error Rate (BER). Hamming Code have good performance at the level modulation of 4-QAM, 8-QAM and 16-QAM. The high-level modulation causing higher probability of error, so the level modulation of 32-QAM and 64-QAM performance less than the maximum because the correction capability is limited to only a single bit error in each block.*

**Keywords:** *Hamming Code, M-QAM, WARP*

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Alhamdulillah segala puji syukur penulis panjatkan kehadirat Allah SWT, karena atas limpahan rahmat dan hidayah-Nya dapat menyelesaikan Tugas Akhir yang berjudul **”Implementasi dan Evaluasi Kinerja Encoder-Decoder Hamming pada *M*-ary Quadrature Amplitude Modulation (M-QAM) Menggunakan Wireless Open-Access Research Platform (WARP)”** dengan lancar.

Tugas akhir ini disusun sebagai salah satu syarat yang harus dipenuhi dalam menyelesaikan program studi Strata 1 (S1) pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember. Selama proses pengerjaan penelitian ini penulis banyak mendapat saran, dorongan, bimbingan serta bantuan dari berbagai pihak. Oleh karena itu dengan segala hormat dan kerendahan hati penulis mengucapkan terima kasih kepada:

1. Kedua orang tua dan adek tercinta di Ponorogo, yang senantiasa mendoakan dan memberikan dukungan dengan segala upaya agar penulis dapat menyelesaikan pendidikan di Perguruan Tinggi.
2. Bapak Dr. Ir. Suwadi dan Ibu Dr. Ir. Titiek Suryani, MT, yang telah memberikan bimbingan selama penelitian berlangsung.
3. Teman-teman satu team WARP yang selalu kompak membantu dan memberikan semangat hingga terselesaikannya Tugas Akhir ini
4. Semua pihak yang telah memberikan dukungan, yang tidak mungkin dapat disebutkan satu persatu dalam laporan ini.

Penulis menyadari bahwa penulisan laporan penelitian ini masih jauh dari sempurna. Oleh karenanya kritik dan saran yang membangun sangat diperlukan untuk perbaikan dan pengembangan lebih lanjut Akhir kata, penulis ucapkan terima kasih.

Surabaya, Juli 2015

Penulis



*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR .....	vii
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Metodologi Penelitian .....	2
1.6 Sistematika Penulisan .....	5
1.7 Manfaat .....	5
BAB II TEORI PENUNJANG .....	7
2.1 Sistem <i>M-ary Quadrature Amplitude Modulation</i> (M-QAM) .....	7
2.1.1 Penentuan Amplitudo Sinyal .....	8
2.1.2 Konstelasi Sinyal .....	9
2.1.3 Pengubahan Sinyal Analog ke Sinyal Digital .....	11
2.1.4 Probabilitas <i>Error Bit</i> .....	13
2.2 Filter <i>Square Root Raised Cosine</i> .....	14
2.3 Estimasi Kanal .....	16
2.4 Kode Hamming .....	17
2.2.1 Encoding Hamming .....	17
2.2.2 Decoding Hamming .....	19
2.2.3 <i>Detection Error dan Correction Error</i> .....	20
2.5 <i>Wireless Open-Access Research Platform</i> (WARP) .....	23
2.3.1 WARP Board .....	24
2.3.2 WARP Lab .....	26
BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM .....	29
3.1 Pemodelan Sistem Komunikasi .....	29
3.2 Implementasi pada Perangkat WARP .....	40
3.3 Metode Pengukuran .....	42
3.3.1 Level Modulasi .....	42
3.3.2 Ukuran code rate Hamming .....	43
3.3.3 Jarak Transmitter dan Receiver .....	43
3.3.4 Perhitungan Eb/No .....	43

BAB IV HASIL DAN ANALISA DATA.....	47
4.1    Modulasi M-QAM Dengan dan Tanpa Kode Hamming .....	47
4.2    Konstelasi Simbol pada Penerima .....	50
4.3    Perbandingan BER dalam Variasi Level Modulasi .....	54
4.4    Perbandingan BER dalam Variasi Ukuran Code Rate .....	58
4.5    Perbandingan BER dalam Variasi Jarak .....	64
BAB V PENUTUP .....	71
5.1    Kesimpulan .....	71
5.2    Saran .....	71
DAFTAR PUSTAKA.....	73

## DAFTAR GAMBAR

Gambar 1. 1 Diagram Alir Metodologi Penelitian .....	3
Gambar 2. 1 Proses Modulasi dan Demodulasi M-QAM [4].....	7
Gambar 2. 2 Konstelasi Sinyal 4-QAM .....	9
Gambar 2. 3 Konstelasi Sinyal 8-QAM .....	10
Gambar 2. 4 Konstelasi Sinyal 16-QAM .....	10
Gambar 2. 5 Konstelasi Sinyal 32-QAM .....	10
Gambar 2. 6 Konstelasi Sinyal 64-QAM .....	11
Gambar 2. 7 Representasi Sudut Sinyal dalam Konstelasi <i>Gray Mapping</i> [4] .....	13
Gambar 2. 8 <i>Transfer Function</i> Sistem.....	15
Gambar 2. 9 <i>Response Impulse</i> Sistem .....	15
Gambar 2. 10 Penyusunan Simbol Pilot .....	16
Gambar 2. 11 Arsitektur WARP [1] .....	23
Gambar 2. 12 Diagram Blok WARP <i>Node</i> [1].....	25
Gambar 2. 13 <i>WARPLap Frame Work</i> [1] .....	27
Gambar 2. 14 <i>WARPLab Flow</i> [3].....	28
Gambar 3.1 Blok Diagram Sistem Komunikasi M-QAM menggunakan Kode Hamming .....	29
Gambar 3. 2 Hasil Pembangkitan Bit dan Encoding .....	30
Gambar 3. 3 Hasil Konversi Bit menjadi Simbol Informasi .....	31
Gambar 3. 4 Scatter Plot Bit informasi yang Dikirim dan Diterima .....	36
Gambar 3.5 Desain Frame.....	37
Gambar 3. 6 Hasil <i>Decoding</i> Hamming (7,4).....	39
Gambar 3. 7 Konfigurasi modul WARP .....	40
Gambar 3. 8 Tx Output Power vs Gain Setting pada WARP[1] .....	41
Gambar 3. 9 Pengukuran menggunakan modul WARP .....	42
Gambar 3. 10 Konstelasi sinyal 8-QAM.....	43
Gambar 3. 11 Konstelasi sinyal 8-QAM digeser ke sisi positif .....	44
Gambar 3. 12 Penggabungan sinyal menjadi satu titik .....	45
Gambar 4. 1 Grafik BER Hasil Simulasi Modulasi M-QAM .....	49
Gambar 4. 2 Konstelasi Sinyal M-QAM (M-ary 4, 8, 16, 32, dan 64) dengan variasi daya pancar .....	51
Gambar 4. 3 Konstelasi sinyal dengan variasi nilai amplitudo pilot .....	53
Gambar 4. 4 Konstelasi yang mengalami pergeseran fasa .....	53
Gambar 4. 5 Grafik BER Terhadap Daya Pancar 1 .....	57
Gambar 4. 6 Grafik BER Terhadap Daya Pancar 2 .....	61
Gambar 4. 7 Grafik BER Terhadap Eb/No .....	64
Gambar 4. 8 BER Terhadap Daya Pancar pada Jarak 3 dan 6 Meter ....	69

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2. 1 Polinomial Galois Field (2).....	18
Tabel 3. 1 Parameter Sistem Komunikasi SISO pada WARP .....	32
Tabel 4. 1 Hasil Pengukuran Tanpa Kode Hamming Jarak 6 M.....	54
Tabel 4. 2 Hasil Pengukuran dengan Kode Hamming (7,4) Jarak 6 M. .	55
Tabel 4. 3 Hasil Pengukuran dengan Kode Hamming (15,11) Jarak 6 M.....	55
Tabel 4. 4 Hasil Pengukuran dengan Kode Hamming (31,26) Jarak 6 M.....	56
Tabel 4. 5 Gain Code untuk Jarak 6 Meter .....	61
Tabel 4. 6 Hasil Pengukuran Tanpa Kode Hamming Jarak 3 M.....	65
Tabel 4. 7 Hasil Pengukuran dengan Kode Hamming(7,4) Jarak 3 M..	65
Tabel 4. 8 Hasil Pengukuran dengan Kode Hamming(15,11) Jarak 3 M.....	66
Tabel 4. 9 Hasil Pengukuran dengan Kode Hamming(31, 26) Jarak 3 M.....	66

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2. 1 Polinomial Galois Field (2).....	18
Tabel 3. 1 Parameter Sistem Komunikasi SISO pada WARP .....	32
Tabel 4. 1 Hasil Pengukuran Tanpa Kode Hamming Jarak 6 M.....	54
Tabel 4. 2 Hasil Pengukuran dengan Kode Hamming (7,4) Jarak 6 M. .	55
Tabel 4. 3 Hasil Pengukuran dengan Kode Hamming (15,11) Jarak 6 M.....	55
Tabel 4. 4 Hasil Pengukuran dengan Kode Hamming (31,26) Jarak 6 M.....	56
Tabel 4. 5 Gain Code untuk Jarak 6 Meter .....	61
Tabel 4. 6 Hasil Pengukuran Tanpa Kode Hamming Jarak 3 M.....	65
Tabel 4. 7 Hasil Pengukuran dengan Kode Hamming(7,4) Jarak 3 M..	65
Tabel 4. 8 Hasil Pengukuran dengan Kode Hamming(15,11) Jarak 3 M.....	66
Tabel 4. 9 Hasil Pengukuran dengan Kode Hamming(31, 26) Jarak 3 M.....	66



*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 1. 1 Diagram Alir Metodologi Penelitian .....	3
Gambar 2. 1 Proses Modulasi dan Demodulasi M-QAM [4].....	7
Gambar 2. 2 Konstelasi Sinyal 4-QAM .....	9
Gambar 2. 3 Konstelasi Sinyal 8-QAM .....	10
Gambar 2. 4 Konstelasi Sinyal 16-QAM .....	10
Gambar 2. 5 Konstelasi Sinyal 32-QAM .....	10
Gambar 2. 6 Konstelasi Sinyal 64-QAM .....	11
Gambar 2. 7 Representasi Sudut Sinyal dalam Konstelasi <i>Gray Mapping</i> [4] .....	13
Gambar 2. 8 <i>Transfer Function</i> Sistem.....	15
Gambar 2. 9 <i>Response Impulse</i> Sistem .....	15
Gambar 2. 10 Penyusunan Simbol Pilot .....	16
Gambar 2. 11 Arsitektur WARP [1] .....	23
Gambar 2. 12 Diagram Blok WARP <i>Node</i> [1].....	25
Gambar 2. 13 <i>WARPLap Frame Work</i> [1] .....	27
Gambar 2. 14 <i>WARPLab Flow</i> [3].....	28
Gambar 3.1 Blok Diagram Sistem Komunikasi M-QAM menggunakan Kode Hamming .....	29
Gambar 3. 2 Hasil Pembangkitan Bit dan Encoding .....	30
Gambar 3. 3 Hasil Konversi Bit menjadi Simbol Informasi .....	31
Gambar 3. 4 Scatter Plot Bit informasi yang Dikirim dan Diterima .....	36
Gambar 3.5 Desain Frame.....	37
Gambar 3. 6 Hasil <i>Decoding</i> Hamming (7,4).....	39
Gambar 3. 7 Konfigurasi modul WARP .....	40
Gambar 3. 8 Tx Output Power vs Gain Setting pada WARP[1] .....	41
Gambar 3. 9 Pengukuran menggunakan modul WARP .....	42
Gambar 3. 10 Konstelasi sinyal 8-QAM.....	43
Gambar 3. 11 Konstelasi sinyal 8-QAM digeser ke sisi positif .....	44
Gambar 3. 12 Penggabungan sinyal menjadi satu titik .....	45
Gambar 4. 1 Grafik BER Hasil Simulasi Modulasi M-QAM .....	49
Gambar 4. 2 Konstelasi Sinyal M-QAM (M-ary 4, 8, 16, 32, dan 64) dengan variasi daya pancar .....	51
Gambar 4. 3 Konstelasi sinyal dengan variasi nilai amplitudo pilot .....	53
Gambar 4. 4 Konstelasi yang mengalami pergeseran fasa .....	53
Gambar 4. 5 Grafik BER Terhadap Daya Pancar 1 .....	57
Gambar 4. 6 Grafik BER Terhadap Daya Pancar 2 .....	61
Gambar 4. 7 Grafik BER Terhadap Eb/No .....	64
Gambar 4. 8 BER Terhadap Daya Pancar pada Jarak 3 dan 6 Meter ....	69

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pada sistem komunikasi digital diperlukan suatu sistem komunikasi yang handal. Salah satu cirinya ialah sistem yang digunakan tahan terhadap gangguan (*noise*). Karena *noise* yang timbul pada saat proses transmisi menyebabkan terjadinya kesalahan bit (*error bit*). Sehingga tidak semua bit yang dikirimkan dapat diterima dengan benar pada sisi penerima. Selain itu, penggunaan modulasi multilevel juga berpengaruh terhadap tingkat kesalahan di penerima. Semakin tinggi level modulasi maka tingkat kesalahannya pun semakin besar.

Oleh karena itu diperlukan suatu upaya untuk memperbaiki kinerja dari sistem. Salah satu upaya yang dilakukan adalah dengan menerapkan suatu algoritma yang dapat mendeteksi kesalahan (*error detection*) sekaligus memperbaiki kesalahan (*error correction*) bit atau yang biasa disebut *error control coding*. Penggunaan *error control coding* juga dapat menambah efisiensi bandwidth dan waktu yang digunakan untuk proses transmisi data. Karena apabila terjadi kesalahan bit yang diterima maka tidak perlu dilakukan pengiriman ulang.

Kode Hamming merupakan salah satu jenis *error control coding* sederhana yang mampu mengoreksi kesalahan bit tunggal dalam satu blok informasi. Untuk mengetahui pengaruh kode Hamming terhadap kehandalan sistem komunikasi digital khususnya pada modulasi multilevel maka, digunakan *M-ary Quadrature Amplitude Modulation* (M-QAM) sebagai objek penelitian. M-QAM adalah salah satu teknik modulasi digital multilevel yang sering digunakan, karena mempunyai konstelasi sinyal tidak begitu rapat dibandingkan dengan modulasi multilevel dari *Quadrature Phase Shift Keying* (QPSK). Secara keseluruhan sistem diimplementasikan dengan menggunakan modul *Wireless Open-Access Research Platform* (WARP). WARP merupakan salah satu *Software Define Radio* (SDR) untuk sistem komunikasi nirkabel yang dapat digunakan sebagai *prototype* jaringan *wireless* dan dapat diimplementasikan secara nyata dan *real-time*. Karena sifatnya yang *programmable*, sangat memungkinkan pengguna untuk lebih mudah dalam melakukan pengembangan sistem tanpa perlu melakukan pergantian *hardware*. Pada penelitian ini modul WARP digunakan sebagai media pemancar dan penerima pada sistem komunikasi digital yang akan diuji. Parameter uji yang digunakan untuk mengukur dan mengevaluasi kinerja kode Hamming ditinjau berdasarkan nilai *Bit Error Rate* (BER) terhadap variasi ( $E_b/N_0$ ) untuk simulasi dan variasi

daya pancar pengirim untuk pengukuran menggunakan WARP pada setiap variasi nilai  $M$ -ary QAM.

## 1.2 Perumusan Masalah

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

1. Bagaimana perbedaan M-QAM tanpa dan dengan kode Hamming yang diimplementasikan menggunakan modul WARP.
2. Berapa kombinasi nilai  $(n, k)$  pada kode Hamming yang mampu memperbaiki kinerja M-QAM dengan nilai  $M$  tertentu.

## 1.3 Batasan Masalah

Untuk menyelesaikan permasalahan-permasalahan yang ada, maka dalam prakteknya akan dibatasi sebagai berikut:

1. Pengkodean yang digunakan yaitu kode Hamming  $(7,4)$ ,  $(15,11)$  dan  $(31,25)$
2. Modulasi yang digunakan yaitu M-QAM
3. Menggunakan modul WARP
4. Menggunakan *software* Matlab
5. Analisa berdasarkan nilai *Bit Error Rate* (BER) yang didapatkan dari hasil pengukuran
6. Pengukuran dilakukan secara *indoor*

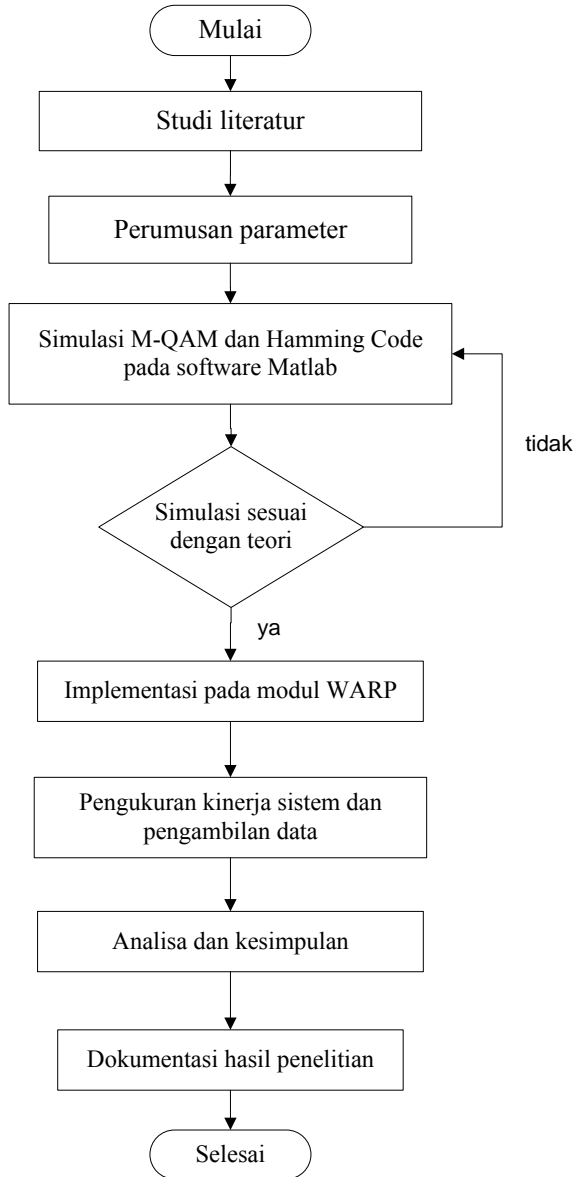
## 1.4 Tujuan

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

1. Mengetahui perbedaan M-QAM tanpa dan dengan kode Hamming yang diimplementasikan menggunakan modul WARP.
2. Mengetahui kombinasi nilai  $(n, k)$  pada kode Hamming yang mampu memperbaiki kinerja M-QAM dengan nilai  $M$  tertentu.

## 1.5 Metodologi Penelitian

Metodologi yang digunakan dalam pengerjaan Tugas Akhir ini dijabarkan dengan diagram alir yang diperlihatkan pada Gambar 1.1:



**Gambar 1. 1** Diagram Alir Metodologi Penelitian

1. Studi Literatur  
Mengumpulkan literatur yang berhubungan dengan topik Tugas Akhir, yaitu :
  - a. Sesuai tema yang digunakan dalam penelitian: Pengolahan sinyal digital, Sistem komunikasi digital, Kode Hamming, *M-ary Quadrature Amplitude Modulation* (M-QAM)
  - b. Simulasi sistem komunikasi
  - c. Desain teknik modulasi digital beserta pengkodean kanal
  - d. Referensi penggunaan modul WARP
  - e. Referensi pemrograman WARPLab
2. Perumusan parameter yang digunakan dan metode pengukuran  
Sebelum proses pengukuran bisa berjalan, parameter performansi yang menunjukkan kualitas unjuk kerja sistem komunikasi dirumuskan dan ditentukan metode pengukurannya. Diantaranya yaitu band frekuensi pengukuran, informasi yang akan dipancarkan, jumlah bit yang akan ditransmisikan, kondisi frame yang dipancarkan serta proses pengukuran kanal. Parameter yang diambil sebagai acuan performansi adalah *Bit Error Rate* (BER).
3. Simulasi sistem komunikasi  
Proses simulasi dilakukan dengan menggunakan software Matlab, yaitu mensimulasikan proses modulasi yang menggunakan penambahan algoritma kode Hamming. Simulasi bertujuan untuk melihat apakah algoritma yang digunakan sesuai dengan teori sistem komunikasi.
4. Implementasi pada modul WARP  
Untuk implementasi pada modul WARP dilakukan dengan menggunakan software WARPlab.
5. Pengukuran data  
Pada pengukuran data dilakukan dengan dua cara yaitu modulasi tanpa dan menggunakan kode Hamming dengan mengubah-ubah *code rate*-nya.
6. Proses analisa data dan pengambilan kesimpulan  
Analisa data dan pengambilan kesimpulan dilakukan setelah pengukuran selesai.
7. Dokumentasi penelitian  
Pada dokumentasi penelitian ini dilakukan pembuatan laporan tugas akhir dan penulisan jurnal ilmiah.

## 1.6 Sistematika Penulisan

Sistematika penulisan buku laporan dari Tugas Akhir ini dijabarkan sebagai berikut:

### 1. BAB I PENDAHULUAN

Bab ini berisi mengenai gambaran permasalahan yang dihadapi dalam penelitian ini, yang mencakup latar belakang, perumusan dan batasan masalah, tujuan, metodologi, sistematika penulisan serta manfaat Tugas Akhir.

### 2. BAB II TEORI PENUNJANG

Bab ini berisi mengenai teori dasar yang dijadikan sebagai acuan yang digunakan dalam penelitian.

### 3. BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM

Bab ini berisi mengenai perancangan untuk pemodelan simulasi dan implementasi dari penelitian yang dilakukan.

### 4. BAB IV HASIL PENGUJIAN DAN ANALISIS

Bab ini berisi mengenai hasil implementasi dari BAB III, yang kemudian dianalisis dan dievaluasi berdasarkan grafik dari hasil pengujian sistem.

### 5. BAB V PENUTUP

Bab ini berisi mengenai kesimpulan dan saran yang diambil berdasarkan dari analisis hasil implementasi.

## 1.7 Manfaat

Manfaat yang diharapkan tercapai setelah selesainya Tugas Akhir ini adalah sebagai berikut:

1. Memahami proses dan cara kerja kode Hamming pada M-QAM dengan menggunakan kanal *real-time*.
2. Memberikan kontribusi tentang proses desain dan implementasi sistem komunikasi pada WARP.
3. Menjadi acuan untuk pengembangan penelitian lebih lanjut khususnya mengenai *error control coding* untuk M-QAM pada WARP.

*Halaman ini sengaja dikosongkan*

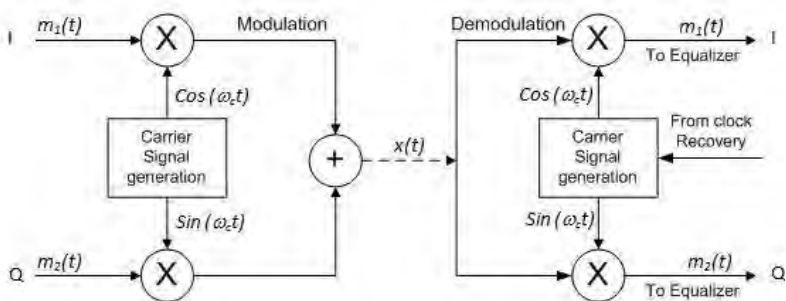


## BAB II TEORI PENUNJANG

### 2.1 Sistem *M*-ary Quadrature Amplitude Modulation (M-QAM)

*M*-ary Quadrature Amplitude Modulation (M-QAM) merupakan salah satu teknik modulasi digital multilevel yang menggabungkan antara modulasi digital jenis *Amplitude Shift Keying* (ASK) dengan *Phase Shift Keying* (PSK), yaitu menyatakan data biner digital 0 dan 1 ke dalam bentuk amplitudo dan fase dari sinyal analog. Sehingga konstelasi sinyalnya berubah sesuai *amplitude* dan juga berdasarkan *phase*.

Blok diagram yang menjelaskan proses modulasi dan demodulasi M-QAM ditunjukkan pada Gambar 2.1 sebagai berikut:



**Gambar 2. 1** Proses Modulasi dan Demodulasi M-QAM [4]

Proses modulasi secara umum dapat dilihat pada Gambar 2.1. Pada umumnya sistem modulasi dibagi menjadi dua yaitu modulator yang berfungsi sebagai *transmitter* dan demodulator sebagai *receiver*. Sinyal informasi yang akan dikirimkan (baseband) dibagi menjadi dua bentuk, *Inphase* merupakan bagian real dan *Quadrature* merupakan bagian imajiner dari sinyal modulasi yang berbeda  $90^\circ$ . Data yang dikirim berupa bit informasi akan dikonversi menjadi sebuah simbol sesuai dengan level modulasi (*M*-ary) yang digunakan. Sehingga didapatkan sejumlah simbol yang akan dipetakan sesuai amplitudo dan fasanya.

Sinyal informasi yang sudah ditransmisikan harus dilakukan proses demodulasi agar didapatkan sinyal informasi sesuai yang

dikirimkan oleh pemancar. Penerima melakukan proses kebalikan dari pemancar. Dengan melewati sinyal pada *matched filter*, maka sinyal yang berfrekuensi tinggi akan dihilangkan sehingga diperoleh sinyal *Inphase*. Begitu pula sinyal *Quadrature*, untuk mendapatkan nilainya diperoleh dengan cara mengalikan sinyal hasil modulasi dengan gelombang sinus. Kedua sinyal tersebut kemudian digabungkan kembali sama seperti bentuk sinyal informasinya.

### 2.1.1 Penentuan Amplitudo Sinyal

Pada proses modulasi M-QAM, aliran bit-bit pertama kali dipetakan dengan *Gray Mapping (Gray Coding)* ke dalam koordinat konstelasi sinyal biner *In-Phase (I)* dan *Quadrature (Q)*, sedangkan sebaliknya pada proses demodulasi M-QAM, aliran sinyal-sinyal analog dikembalikan menjadi sinyal-sinyal biner dengan menggunakan *Gray Mapping (Gray Coding)*. [4]

Sinyal hasil modulasi M-QAM dapat dinyatakan ke dalam persamaan 2.1:

$$S_m(t) = A_{m_I} \cos \theta + j A_{m_Q} \sin \theta \quad (2.1)$$

Dimana sudut dan phase dapat diperoleh dari:

$$\theta = \omega_c t = 2\pi f_c t \quad (2.2)$$

Dan amplitudo sinyal ( $A_m$ ) dapat dinyatakan sebagai:

$$A_m = \sqrt{\frac{2 \cdot E_m(t)}{T}} \quad (2.3)$$

Dengan  $E_m(t)$  adalah energi sinyal,  $m = 1, 2, \dots, M$ , dan  $0 \leq t \leq T$ .

$M$ -ary ( $M$ ) adalah jumlah multilevel modulasi atau jumlah maksimum data pada M-QAM dalam bentuk untaian bit. Untuk sistem modulasi 4-QAM dapat dikirimkan 4 data dalam biner, sedangkan untuk 64-QAM dapat dikirimkan 64 data dalam biner. Jumlah untaian biner yang dapat merepresentasikan data disebut dengan ukuran konstelasi bit, disimbolkan dengan  $k$ , dan dapat dinyatakan sebagai:

$$k = \log_2 M \quad (2.4)$$

dengan  $A_m$  selain sebagai amplitudo sinyal, juga dapat merepresentasikan jumlah maksimum data yang dikirimkan mengikuti persamaan:

$$A_m = M - 1 \quad (2.5)$$

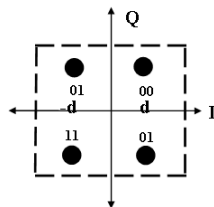
Misalnya, pada sistem modulasi 16-QAM, amplitudo sinyal atau jumlah data yang dapat dikirimkan sebanyak 4 data, yang dimulai dari 0 hingga mencapai nilai maksimum:

$$M - 1 = 16 - 1 = 15$$

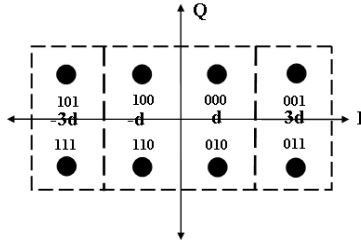
### 2.1.2 Konstelasi Sinyal

Pemetaan sinyal pada modulasi M-QAM dengan *Gray Mapping* (*Gray Coding*) dapat dilakukan dengan mengkombinasikan sisi tegak (*Quadrature = Q*) dan sisi sudut (In-Phase=I) konstelasi sinyal. Angka-angka yang secara umum digunakan untuk menyatakan nilai  $A_{mQ}$  sebagai sisi tegak Q dan  $A_{mI}$  sebagai sisi sudut I adalah konstanta d.

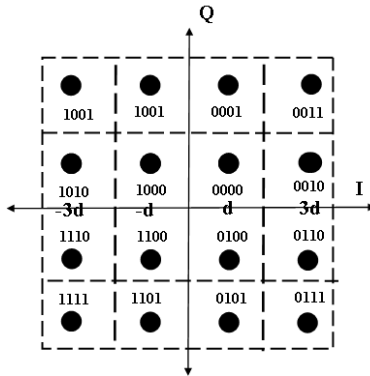
Untuk modulasi 4-QAM, terdapat pasangan (+d, -d) dimana nilai d menyatakan bit 0 dan -d menyatakan bit 1. Untuk 8-QAM, pada sisi sudut I ditambahkan nilai (-3d, +3d), dimana pada penambahan titik -3d dan +3d, ditambahkan bit 1, sedangkan pada -d dan +d, ditambahkan bit 0. Untuk 16-QAM, pada sisi tegak Q dan sisi sudut I ditambahkan nilai (-3d, +3d), dimana pada penambahan titik -3d dan +3d di sisi tegak Q, ditambahkan bit 1, sedangkan pada -d dan +d, ditambahkan bit 0. Untuk 32-QAM, pada sisi sudut I ditambahkan nilai (-5d, +5d) dan (-7d, +7d), sedangkan pada 64-QAM, pada sisi sudut I dan sisi tegak Q ditambahkan nilai (-5d, +5d) dan (-7d, +7d), seperti ditunjukkan pada gambar sebagai berikut:



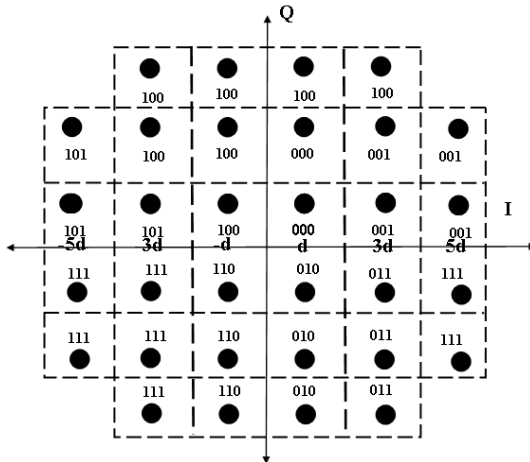
**Gambar 2. 2** Konstelasi Sinyal 4-QAM



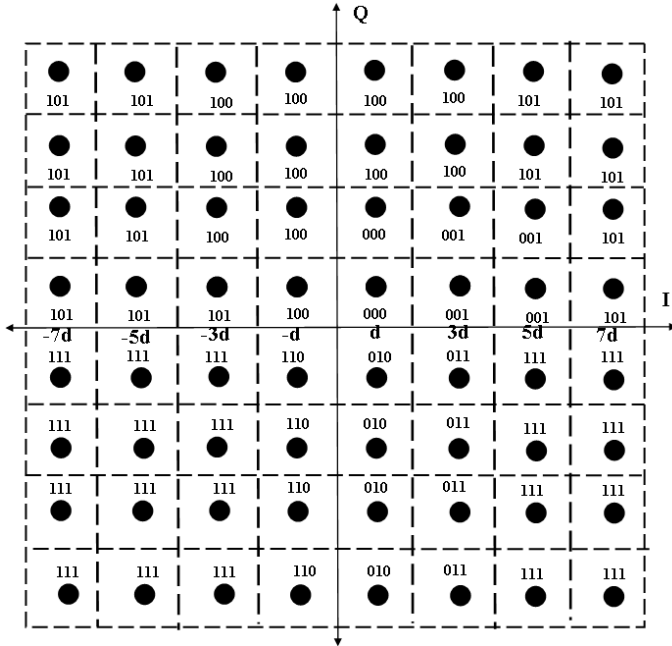
**Gambar 2. 3** Konstelasi Sinyal 8-QAM



**Gambar 2. 4** Konstelasi Sinyal 16-QAM



**Gambar 2. 5** Konstelasi Sinyal 32-QAM



Gambar 2. 6 Konstelasi Sinyal 64-QAM

### 2.1.3 Pengubahan Sinyal Analog ke Sinyal Digital

Pasangan konstelasi sinyal dalam *Gray Mapping (Gray Coding)*, yaitu sisi tegak (*Quadrature = Q*) dan sisi sudut (*In-phase = I*), sehingga dapat dituliskan pada persamaan berikut [4]:

$$Am_I = \sqrt{\frac{2E_m(t)}{T}} \cos(2 f_c t) \quad (2.6)$$

$$Am_Q = \sqrt{\frac{2E_m(t)}{T}} \sin(2 f_c t) \quad (2.7)$$

Dimana untuk  $A_{mI}$  dan  $A_{mQ}$  dapat dinyatakan dalam jarak (distance) koordinat :

$$(I, Q) = \pm(2n - 1)d \quad (2.8)$$

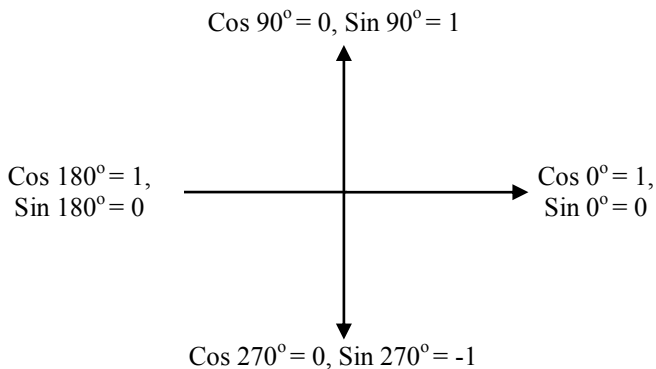
dengan  $n = 1, 2, 3, \dots, n$   
 dengan nilai  $d$  dapat dicari dengan persamaan:

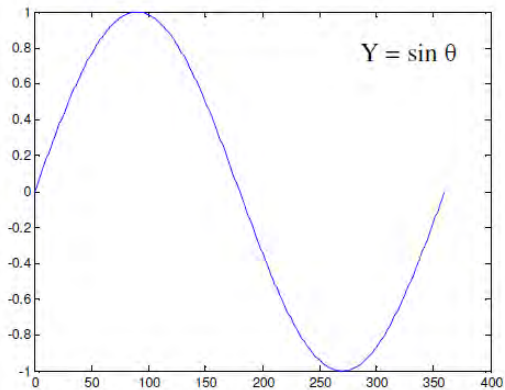
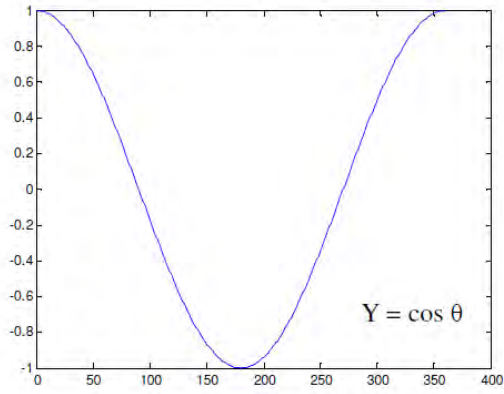
$$d = \sqrt{\frac{4E_m}{A_{mI}^2 + A_{mQ}^2}} = \sqrt{\frac{E}{10}} \quad (2.9)$$

Sinyal-sinyal pada M-QAM dapat dinyatakan dengan  $(I, Q)$ . Untuk 4-QAM, sinyal-sinyal dimulai dari  $(+d, +d)$ , yaitu :00 (0) hingga  $(-d, -d)$ , yaitu :11 (3). Sedangkan untuk 64-QAM, sinyal-sinyal dimulai dari  $(+7d, +7d)$ , yaitu :000000 (0) hingga  $(-7d, -7d)$ , yaitu :111111 (63). Konstanta  $d$  dapat digunakan sebagai deret untuk mempresentasikan angka biner ke amplitudo ( $A$ ) dan sudut ( $\theta$ ) sinyal sebagai berikut:

$$D = \frac{k}{M} \sum_{i=1}^N (2n_i - 1) \cdot d \quad (2.10)$$

Sedangkan untuk menyatakan sudut ( $\theta$ ), pasangan koordinat  $(I, Q)$  disesuaikan dengan penempatannya pada *Gray Mapping (Gray Code)* dapat dijelaskan pada gambar sebagai berikut:





**Gambar 2. 7** Representasi Sudut Sinyal dalam Konstelasi *Gray Mapping* [4]

### 2.1.4 Probabilitas *Error Bit*

Dalam penerimaan bit informasi, pada bagian penerima memungkinkan terjadinya kesalahan atau biasa yang disebut *probabilitas error bit* ( $P_b$ ). Untuk menunjukkan kinerja sistem, digunakan probabilitas error yang dijadikan acuan untuk mendapatkan nilai *Bit Error Rate* (BER). Nilai BER diperoleh

dengan membandingkan jumlah bit salah yang di terima dengan jumlah bit yang dikirimkan. Semakin besar nilai BER maka kualitas dari proses transmisi tersebut semakin jelek. Pada konstelasi *rectangular* dengan kondisi kanal *Gaussian* serta penerima *matched filter*,  $P_b$  untuk M-QAM dinyatakan dalam persamaan 2.11 :

$$P_b = \frac{2(1 - L^{-1})}{\log_2 L} Q \left[ \sqrt{\frac{3 \log_2 L 2 E_b}{M - 1 N_o}} \right] \quad (2.11)$$

dimana  $M = 2^k$  dan  $k = 2, 4, 6, \dots, n$  (bilangan genap)

$Q(x)$  merupakan *Q-function* yang disebut *complementary error function* atau *co-error function*, didefinisikan sebagai *probabilitas error* simbol pada *noise Gaussian* dan  $L = \sqrt{M}$  yang merepresentasikan level amplitudo pada satu dimensi. Sedangkan perbandingan nilai energi bit dengan *noise* ( $E_b/N_o$ ) didapatkan dari persamaan 2.12 :

$$\frac{E_b}{N_o} = \frac{S W}{N R} \quad (2.12)$$

Dimana  $S$  = Daya sinyal

$N$  = Daya *noise* yang terukur

$W$  = Bandwidth

$R$  = Laju bit yang digunakan

## 2.2 Filter *Square Root Raised Cosine*

*Square Root Raised Cosine* adalah salah satu filter digital yang berfungsi untuk meminimalkan *inter symbol interference* (ISI). ISI yang timbul disebabkan karena kapasitas kanal yang terbatas sehingga pada saat pulsa-pulsa kotak dilewatkan, akan menyebar dalam domain waktu. Pulsa untuk setiap simbol akan terdistorsi dalam interval waktu dari urutan simbol yang berturut-turut. Sehingga terjadi peningkatan probabilitas *error* pada penerima dalam mendeteksi simbol.

Dalam pemrosesan sinyal, filter *Raised Cosine* sering digunakan sebagai filter dalam pemancar dan penerima dalam sistem komunikasi digital untuk melakukan *matched filtering*. Hal ini membantu dalam meminimalisasi efek ISI. Filter *Raised Cosine* merupakan tipe dari



*Finite Impulse Response* (FIR) yang digunakan sebagai *Low Pass Filter* (LPF). Fungsi transfer pada *Raised Cosine* adalah sebagai berikut:

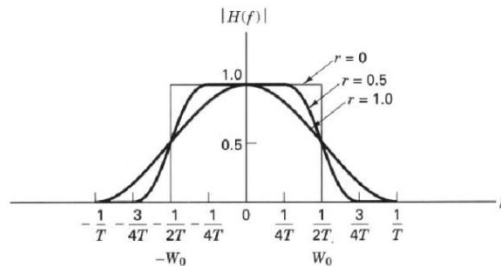
$$H_e(f) = \begin{cases} T & |f| < \frac{1-\beta}{2T} \\ \frac{T}{2} \left\{ 1 + \cos \left( \frac{\pi T}{\beta} \left[ |f| - \frac{1-\beta}{2T} \right] \right) \right\} & \frac{1-\beta}{2T} < |f| < \frac{1+\beta}{2T} \\ 0 & \text{yang lain} \end{cases} \quad (2.13)$$

dimana:

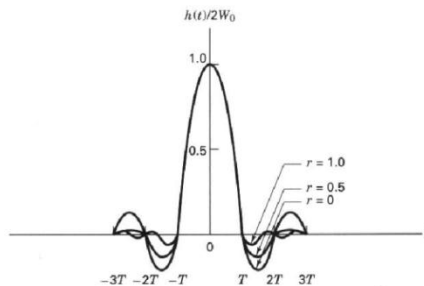
$\beta$  : faktor roll-off

$f$  : frekuensi

Spektrum filter *raised cosine* menunjukkan bentuk yang simetri dengan nilai ganjil  $\frac{1}{2}T$ , dimana T merupakan periode simbol dari sistem. Untuk *transfer function* dan *impulse response* sistem dapat dilihat pada gambar berikut:



**Gambar 2. 8** *Transfer Function* Sistem



**Gambar 2. 9** *Response Impulse* Sistem

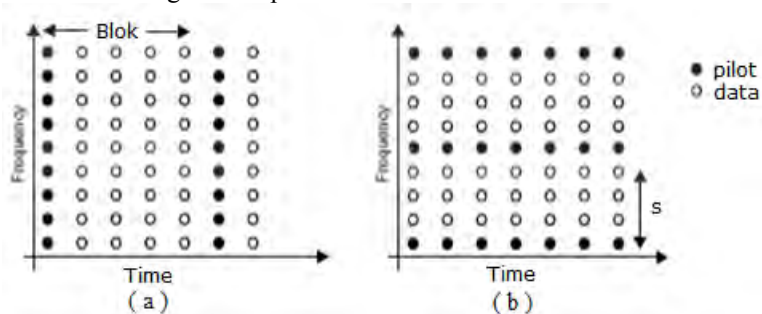
### 2.3 Estimasi Kanal

Estimasi kanal adalah suatu teknik yang digunakan pada transmisi yang bertujuan untuk memperkirakan atau mengestimasi *impulse respon* dari suatu kanal terhadap sinyal terkirim. Efek perubahan terhadap sinyal terkirim yang dihasilkan oleh kanal harus dilakukan estimasi agar pendeteksian sinyal informasi menjadi lebih akurat.

Salah teknik dari estimasi kanal adalah *pilot assisted channel estimation*, bekerja dengan mengirimkan *pilot symbol* atau deretan bit bersamaan dengan informasi bit. Penerima sebelumnya telah mengetahui nilai pilot yang akan dikirimkan pemancar. Dengan pilot yang diketahui tersebut maka ketika ditransmit dan melewati kanal, penerima bisa membandingkan nilai pilot yang sebelumnya telah diketahui dengan nilai pilot yang berubah melewati kanal. Selanjutnya, dengan suatu teknik teknik algoritma estimasi kanal bisa didapatkan *channel impulse response*-nya.

Pilot simbol digunakan untuk menentukan pola perubahan yang terjadi. Selanjutnya dengan pola perubahan ini kita dapat mengetahui impuls respon kanalnya. Dengan metode interpolasi maka sinyal informasi sebelum dilewatkan kanal dapat diperkirakan sehingga error yang terjadi dapat diperkecil. Secara umum, terdapat dua dasar penyusunan simbol pilot yaitu:

- Block type pilot*, yaitu dengan memasukkan simbol pilot kedalam semua subcarrier dalam waktu tertentu dalam periode waktu tertentu. Selain penerima sebelumnya telah mengetahui nilai pilot, penerima juga telah mengetahui kapan pilot serempak dikirim.
- Comb type pilot*, yaitu disediakan alokasi frekuensi khusus yang digunakan untuk mentransmit pilot setiap waktu. Pengirim sebelumnya telah menentukan subcarrier mana yang digunakan untuk mengirimkan pilot simbol.



**Gambar 2. 10** Penyusunan Simbol Pilot

## 2.4 Kode Hamming

Kode Hamming merupakan kode pengoreksi kesalahan tunggal (*single error-correction code*) yang pertama kali diperkenalkan oleh Richard Hamming pada tahun 1950. Metode penanganan error yang digunakan kode Hamming ialah *Forward Error Control*, sebelum data dikirimkan data akan dikodekan dengan suatu pembangkit kode (enkoder) untuk menyisipkan beberapa bit pariti yang kemudian dikirimkan ke penerima. Pada penerima terdapat sebuah penerjemah kode (dekoder) yang mendekodekan data tersebut, dan apabila terjadi *error* pada data akan dilakukan pengoreksian. Sehingga pengirim tidak perlu melakukan pengiriman ulang bila data yang diterima mengalami *error*.

Ide dasar pengkodean Hamming adalah menggunakan *parity-checking*, yaitu menambahkan bit pariti pada blok data (bit-bit informasi). Bit pariti ini berfungsi untuk mendeteksi bit yang salah, sekaligus menentukan lokasi kesalahan bit tersebut. Jika suatu saat ada perubahan bit-bit data ketika proses transmisi, maka bit-bit informasi asli masih bisa diselamatkan. [6]

Kode Hamming memiliki ukuran yang disebut *block code*  $(n, k)$ . Untuk setiap integer positif  $m \geq 3$  terdapat parameter sebagai berikut [6] :

Panjang kode	: $n = 2^m - 1$
Jumlah bit informasi	: $k = 2^m - m - 1$
Jumlah bit pariti	: $m = n - k$
Kapasitas koreksi <i>error</i>	: $t = 1$ ( $d_{min} = 3$ )

Semua kode Hamming mempunyai jarak kode minimum ( $d_{min}$ ) 3, ini berarti bahwa kode hamming mampu mendeteksi dan mengoreksi *single error* dan mendeteksi *double errors*. Umumnya ukuran kode Hamming adalah  $(7,4)$ ,  $(15,11)$  dan  $(31,26)$ , yang kesemuanya mempunyai jarak Hamming yang sama.[7]

### 2.2.1 Encoding Hamming

Algoritma pembentukan kode Hamming dimulai dengan mengambil sebanyak  $k$  bit data yang dikalikan dengan generator matriks. Proses perkalian bit data dengan generator matriks bertujuan untuk menyisipkan bit *parity check*. Jumlah *parity check* untuk masing-masing code rate berbeda-beda. Oleh karena itu dalam pengkodean Kode Hamming diperlukan adanya suatu generator matriks  $G$ . Generator matriks yang dipilih adalah generator matriks yang sistematis. Bentuk

umum generator matriks dari Kode Hamming yang sistematis sesuai persamaan (2.14). [6]

$$G = [P_{kx(n-k)} : I_k] = \begin{bmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,n-k-1} & 1 & 0 & \dots & 0 \\ P_{1,0} & P_{1,1} & \dots & P_{1,n-k-1} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ P_{k-1,0} & P_{k-1,1} & \dots & P_{k-1,n-k-1} & 0 & 0 & \dots & 1 \end{bmatrix} \quad (2.14)$$

Generator matriks untuk Kode Hamming  $(n, k)$  dapat juga dihitung menggunakan primitif polinomial. Langkah awal yang harus dilakukan sebelum menyusun generator matriks adalah menentukan jumlah bit pariti ( $m$ ) yang akan digunakan untuk menghitung nilai  $(n)$  dan  $(k)$  seperti yang telah dijelaskan sebelumnya. Selanjutnya, menentukan nilai polinomial berdasarkan tabel *Galois Field* ( $2^m$ ) atau ( $GF(2^m)$ ) seperti pada tabel 2.1.

**Tabel 2. 1** Polinomial Galois Field (2)

Jumlah bit pariti	Polinomial
3	1101
4	11001
5	101001
6	1100001
7	10010001

Kemudian susun matriks generatornya dengan tahapan sebagai berikut [6] :

1. Gunakan polinomial sebagai baris pertama dan tambahkan nilai 0 hingga jumlah bit seluruhnya menjadi sebanyak  $n$ . Pola bit yang dihasilkan diberi nama sebagai  $g(x)$ .
2. Buat baris kedua dengan menggeser bit terakhir pada baris pertama ke bit pertama, prosesnya adalah  $xg(x)$
3. Buat ketiga dengan menggeser bit terakhir pada baris kedua ke bit pertama, prosesnya adalah  $x^2g(x)$ . Pergeseran bit terakhir terus dilanjutkan sampai sejumlah  $k$  baris. Matriks  $G$  yang terbentuk adalah sebagai berikut:

$$G = \begin{bmatrix} g_0(x) = g(x) \\ g_1(x) = xg(x) \\ \vdots \\ g_{k-1}(x) = x^{k-1}g(x) \end{bmatrix} \quad (2.15)$$

Generator yang terbentuk merupakan generator yang tidak sistematis. Untuk mengubahnya menjadi sistematis yaitu dengan operasi baris elementer. Matriks dari  $G$  yang sistematis merupakan generator dari Kode Hamming.

$$G = \begin{bmatrix} g_0(x) = g(x) \\ g_1(x) = xg(x) \\ \vdots \\ g_{k-1}(x) = xg_{k-2}(x) + g(x) \end{bmatrix} \quad (2.16)$$

*Encoder* kode Hamming akan menghasilkan  $n$  bit *output codeword*, yang terdiri dari  $k$  bit informasi dan  $m$  bit *parity*. Pada proses *encoding* kode Hamming,  $n$  bit *output codeword* dihasilkan dengan mengalikan  $k$  bit matriks informasi dengan generator matriks yang mempunyai ukuran  $n \times k$ . Proses tersebut dapat didefinisikan dengan persamaan berikut:

$$C = [M][G] \quad (2.17)$$

Dimana  $C$  = *Output codeword*;  
 $M$  = Matriks bit informasi;  
 $G$  = Matriks generator kode blok.

### 2.2.2 Decoding Hamming

Decoder kode Hamming berfungsi untuk mendeteksi kesalahan *syndrome error* dan juga memperbaiki kesalahan yang timbul pada *syndrome error* tersebut. Untuk melakukan koreksi kesalahan pada kode hamming, harus diketahui matriks *parity check*-nya yang dinyatakan dengan persamaan:

$$H = [I P^T] \quad (2.18)$$

Dimana  $H$  = matriks cek *parity*  
 $I$  = matriks identitas  
 $P^T$  = hasil transpose dari matriks *parity*  $P$ .

Dimana matriks *parity check* diatas dapat dihitung sindrom dengan persamaan :

$$S = e.H^T \quad (2.19)$$

Dimana  $S$  = bentuk *syndrome error*  
 $e$  = bentuk *error code*  
 $H^T$  = matriks *transpose parity check*

Untuk mengetahui *error* dalam *codeword*, dilakukan dengan membandingkan *syndrome error* dari kode masing-masing jenis hamming dengan *syndrome error codeword* yang diterima. *Syndrome error* diperoleh dengan mengalikan *codeword* yang diterima dengan transpose matriks cek *parity*.

### 2.2.3 *Detection Error dan Correction Error*

Nilai *syndrome error* menentukan ada tidaknya *error* pada *codeword* yang diterima. Bila *syndrome error* bernilai nol, maka kode yang diterima tidak terkena *error*. Sehingga, kode yang dikirimkan ( $r$ ) = kode yang diterima ( $e$ ). Sedangkan bila *syndrome error* tidak bernilai nol, maka ada *error* terdeteksi. Maka langkah yang harus dilakukan adalah sebagai berikut:

1. Mencari posisi *error* berdasarkan posisi ke berapa pola bit *syndrome* tersebut pada kolom matriks *transpose parity check* (HT).
2. Lakukan koreksi bit-bit yang terkena *error*, yakni dengan meng-*invert* nilai dari kode yang diterima (bila 0 menjadi 1, begitu juga sebaliknya), sehingga diperoleh kode yang dikirimkan.
3. Hilangkan semua bit *parity*-nya untuk mendapatkan kembali bit informasi dalam *codeword*.

Misalnya sebuah generator Hamming memiliki *parity* ( $m$ ) = 3, maka:

$$\begin{aligned} n &= 2^m - 1 = 7 \\ k &= 2^m - m - 1 = 4 \\ (n, k) &= (2^m - 1, 2^m - m - 1) = (7, 4) \end{aligned}$$

Sehingga merupakan kode Hamming dengan *block code* (7,4). Berdasarkan tabel Primitive Polynomial GF ( $2^m$ ), untuk ( $m$ ) = 3 diperoleh pola primitif polinomial sebagai berikut:

$$g(x) = 1 + X + X^3 = 1101 \tag{2.20}$$

Maka diperoleh generator matriks yang tidak sistemik:

$$G = \begin{bmatrix} g_0(x) = g(x) \\ g_1(x) = xg(x) \\ g_2(x) = x^2g(x) \\ g_3(x) = x^3g(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Dengan operasi baris elementer. Diperoleh generator matriks  $G$  yang sistemik:

$$G = \begin{bmatrix} g_0(x) = g(x) \\ g_1(x) = xg(x) \\ g_2(x) = xg_1(x) + g(x) \\ g_3(x) = xg_2(x) + g(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Kemudian, misalnya data yang diberikan oleh sumber data adalah  $M = 0101$ . Maka bit *output codeword* yang dihasilkan dari proses *encode* kode Hamming adalah:

$$C = [M][G]$$

$$C = [0 \quad 1 \quad 0 \quad 1] \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C = [1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1]$$

Jadi kode yang dikirimkan ( $r$ ) = 110010. Selanjutnya kode Hamming (7,4) memiliki matriks *parity check* :

$$H = [I P^T]$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Proses dekodenya sebagai berikut:

1. Untuk kanal tanpa *noise*, kode yang diterima ( $e$ ) = [1 1 0 0 1 0 1], maka bentuk *syndrome error* dari kode yang diterima:

$$S = e.H^T$$

$$S = [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$S = [0 \ 0 \ 0]$$

Karena *syndrome error* bernilai nol, berarti kode yang diterima tidak terjadi *error* sehingga tidak perlu dilakukan koreksi. Kode yang dikirimkan ( $r$ ) sama dengan kode yang diterima ( $e$ ) = [1 1 0 0 1 0 1]. Setelah dilakukan pembuangan bit-bit *parity*-nya, diperoleh kembali bit-bit informasi dalam *codeword* (data yang dikirim)  $M = 0101$

- Untuk kanal dengan *noise*, kode yang diterima ( $e$ ) = [1 1 0 0 **0** 0 1], maka bentuk *syndrome error* dari kode yang diterima:

$$S = e.H^T$$

$$S = [1 \ 1 \ 0 \ 0 \ \mathbf{0} \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$S = [0 \ 1 \ 1]$$

Karena *syndrome error* bukan bernilai nol, berarti kode yang diterima terjadi *error*. Sehingga perlu dilakukan pengecekan ulang dengan mencocokkan bit *syndrome error* dengan bit-bit matriks  $H^T$  dan diperoleh kecocokan pada baris ke-5 pada matriks  $H^T$ , dengan demikian kesalahan bit terjadi pada bit ke-5. Oleh karena itu bit ke-5 dari kode yang diterima ( $e$ ) = 1 1 0 0 [**0**] 0 1 yang bernilai 0 diubah menjadi 1, sehingga data yang benar adalah 1 1 0 0 1 0 1. Setelah dilakukan pembuangan bit-bit *parity*-nya, diperoleh kembali bit-bit informasi dalam *codeword* (data yang dikirim)  $M = 0101$ .

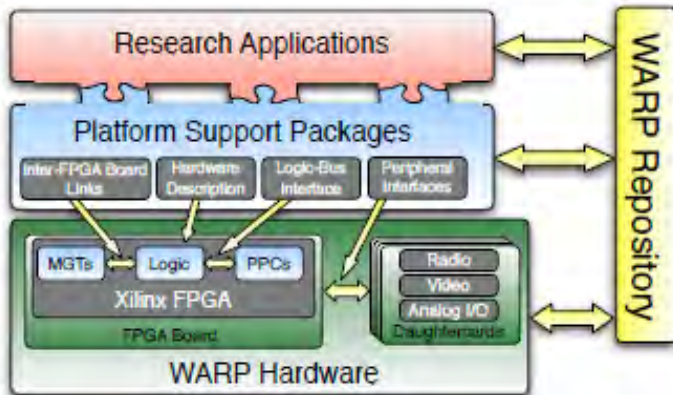
Berdasarkan dua contoh diatas dapat dilihat bahwa kode Hamming mempunyai kemampuan untuk mengoreksi kesalahan.



## 2.5 Wireless Open-Access Research Platform (WARP)

*Wireless Open-Access Research Platform (WARP)* adalah salah satu teknologi *Software Defined Radio (SDR)* yang dikembangkan untuk modul penelitian tentang sistem komunikasi nirkabel. Dimana SDR sendiri merupakan sebuah sistem komunikasi radio yang memanfaatkan perangkat lunak. Sehingga untuk keperluan modifikasi, SDR lebih fleksibel karena tidak perlu melakukan banyak pergantian infrastruktur dari sistem yang sudah ada. Sama halnya dengan WARP, *platform wireless* ini dapat diprogram untuk membuat *prototype* jaringan *wireless* mulai yang dasar hingga kompleks. [1]

Arsitektur WARP terdiri dari empat komponen utama, seperti yang diperlihatkan pada Gambar 2.11 berikut.



**Gambar 2.11** Arsitektur WARP [1]

a. *Custom Hardware*

Perangkat keras yang disesuaikan untuk memenuhi kualifikasi sistem komunikasi dengan kinerja yang tinggi. Perangkat keras ini menyediakan sumber daya untuk *Digital Signal Processing (DSP)* berkecepatan tinggi yang bisa digunakan untuk implementasi algoritma yang intensif, terukur interkoneksi untuk mengalokasikan lebih banyak daya pemrosesan yang diperlukan, prosesor khusus, dan ekspansi *interface* yang lainnya.

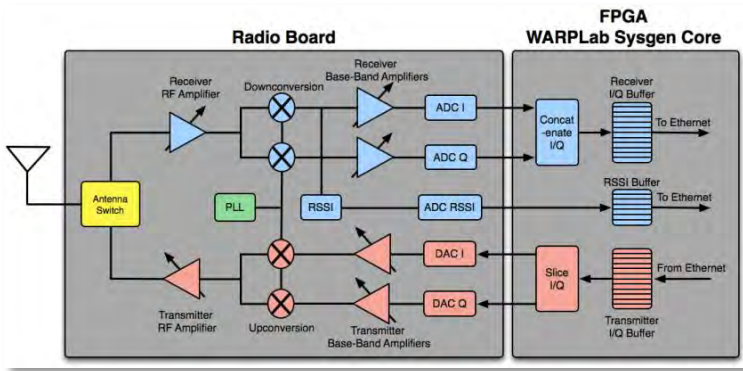
- b. *Platform Support Packages*  
Alat desain dan modul antarmuka tingkat rendah dirancang untuk memungkinkan penggunaan *hardware* oleh para peneliti di seluruh lapisan desain jaringan nirkabel.
- c. *Open-Access Repository*  
Pusat penyimpanan untuk semua *source code*, model dan *file* desain *hardware*. *Content* secara lengkap akan tersedia di bawah lisensi BSD seperti *open-source*.
- d. *Research Applications*  
Implementasi dari algoritma baru, diaktifkan oleh *hardware* khusus dan paket mendukung *platform*. Sistem Kendali menggabungkan algoritma baru dapat dengan cepat dibangun menggunakan bangunan standar modul blok yang disediakan dalam *repository*.

Dalam penggunaan WARP pada penelitian terdapat beberapa metode untuk memprogram perangkat tersebut. 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. Metode kedua dengan membangun sistem *real-time* yang diinginkan untuk diimplementasikan pada FPGA (Field Programmable Gate Array). Sistem tersebut bisa dibangun secara independen dan berdiri pada node WARP.[2]

### 2.3.1 WARP Board

Proses implementasi sistem komunikasi pada WARP dilakukan pada modul WARP sebagai FPGA dan modul radio sebagai antar muka antar sistem komunikasi dengan pengolahan sinyal digital. Proses pengiriman dan penerimaan sinyal informasi, pengolahan sinyal digital dilakukan dengan menggunakan FPGA dan dengan radio board. [2]

Sistem WARP secara umum dijelaskan pada Gambar 2.12 Berdasarkan Gambar 2.12, pada *node* WARP terdapat dua komponen utama, yaitu *Radio Board* dan FPGA. Sementara untuk warna merah muda menandakan blok pemancar dan warna biru menandakan blok penerima.



**Gambar 2. 12** Diagram Blok WARP Node [1]

Blok pemancar:

1. Tx I/Q Buffer  
Tempat penyimpanan sampel *inphase* dan *quadrature* yang akan dikirim.
2. DAC I/Q  
Mengkonversi dari digital ke analog untuk sample *inphase/quadrature*.
3. Tx BB Amplifier (*Transmitter Base Band Amplifier*)  
Terdapat dua amplfier 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.
4. *Upconversion*  
Mengubah sinyal *baseband* ke sinyal RF. Frekuensi *carrier* tergantung pada pengaturan *Phase Locked Loop* (PLL).
5. Tx RF Amplifier  
*Transmitter RF amplifier, Variabel gain amplifier*. Nilai ini yang dimasukkan oleh *user* di *workspace* Matlab.

Blok penerima:

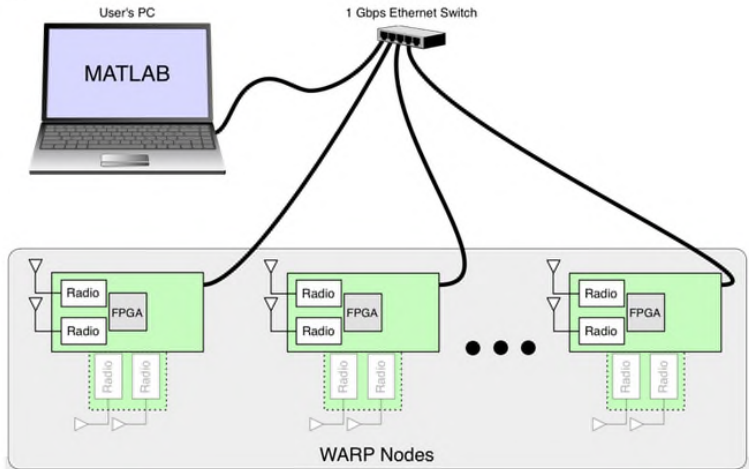
1. Rx RF Amplifier  
*Receiver RF amplifier, variable gain amplifier.* Nilai ini yang dimasukkan oleh *user* di *workspace* Matlab.
2. *Downconversion*  
Mengubah sinyal RF ke sinyal *baseband*. Sinyal *baseband* tersebut terdiri dari sinyal *inphase* dan *quadrature*. *Downconversion* dari RF ini bergantung kepada *setting* PLL.
3. *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.
4. *Receive Signal Strength Indicator (RSSI)*  
Blok ini mengukur nilai dari RSSI.
5. ADC I/Q  
Mengkonversi sinyal analog ke digital untuk sampel *inphase* dan *quadrature*.
6. ADC RSSI  
Mengkonversi sinyal digital ke analog untuk pengukuran RSSI. Data RSSI tersedia di  $\frac{1}{4}$  laju data I/Q.
7. Rx I/Q *Buffer*  
Tempat penyimpanan sample yang diterima baik dalam bentuk *inphase* maupun *quadrature*.
6. RSSI *Buffer*  
Tempat penyimpanan data RSSI.

Pengaturan *Phase Locked Loop* (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 *framework* untuk *prototype physical layer* yang menggabungkan Matlab dengan hardware WARP. Dengan WARPLab, kita dapat secara langsung membangkitkan sinyal dari Matlab dan dapat ditransmisikan secara *real-time* melalui udara menggunakan node WARP. Penggunaan dari WARPLab memungkinkan kombinasi dari beberapa sistem multi-antena pada

pemancar dan penerima. Sehingga pengguna lebih mudah untuk mengaplikasikan bermacam-macam sistem baik *Single Input-Single Output (SISO)*, *Multi Input-Multi Output (MIMO)* ataupun kombinasi dari keduanya. Secara umum arsitektur WARPLab ditunjukkan pada Gambar 2.13. [2]

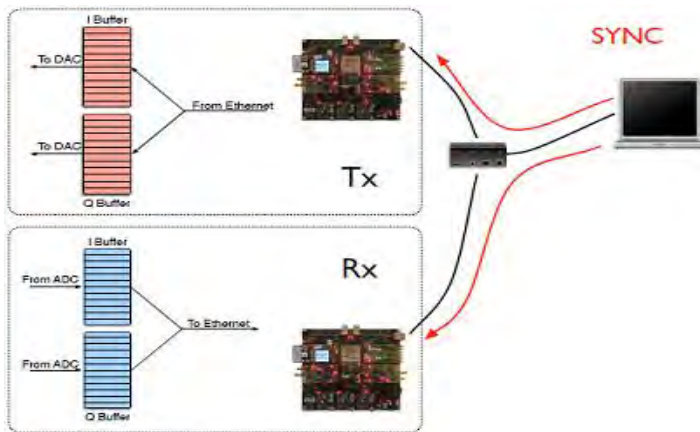


**Gambar 2. 13** *WARPLap Frame Work* [1]

Berikut beberapa informasi penting yang berhubungan dengan *WARPLab Frame Work* [3]:

- Satu PC/laptop dapat mengontrol banyak node-node WARP.
- Matlab untuk pemrosesan sinyal.
- Pemrosesan tidak *real time*.
- WARP untuk *interface wireless*.
- 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.

Sementara untuk aliran kerja dari WARPLab diperlihatkan pada Gambar 2.11 dibawah ini [3]:



**Gambar 2. 14** WARPLab Flow [3]

Alir kerja dari WARPLab meliputi:

- Pengguna membuat konfigurasi dalam Matlab untuk inisialisasi node-node dan setting radio. Sampel yang akan dikirim adalah bagian dari *custom PHY*.
- Sampel yang akan dikirim akan didownload ke *buffer* di *node* yang digunakan sebagai pemancar (Tx).
- Pengguna mengirimkan *trigger* untuk *node transmitter* dan *receiver*. Setelah penerimaan *trigger* ini, sampel dikirim *over-the-air* dan di-*capture* secara *real-time*.
- Pengguna membaca sampel yang ter-*capture* diambil dari *node* penerima dengan *workspace* Matlab.
- Sampel yang diterima diproses secara *offline* di Matlab

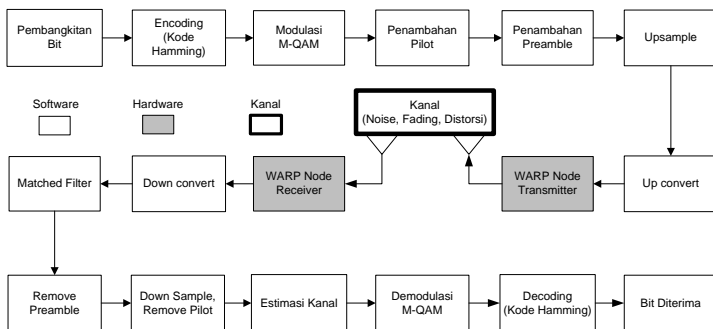
## BAB III

### PERANCANGAN DAN IMPLEMENTASI SISTEM

Untuk mengetahui lebih lanjut mengenai kinerja dari Kode Hamming, maka perlu dilakukan perancangan sistem dan metode pengukuran yang jelas. Seperti yang sudah dijelaskan sebelumnya, teknik modulasi yang digunakan yaitu *M-ary Quadrature Amplitude Modulation* (M-QAM). Pada penelitian ini pemodelan sistem komunikasi menggunakan model *Single Input Single Output* (SISO) yang dirancang melalui *software* Matlab dan diimplementasikan menggunakan modul WARP. Beberapa uraian mengenai perancangan dan implemementasi sistem akan dijelaskan sebagai berikut:

#### 3.1 Pemodelan Sistem Komunikasi

Pemodelan sistem komunikasi SISO dimodelkan melalui dua tahap, pertama pada *software* Matlab sebagai simulasi untuk menguji bahwa sistem yang dirancang telah sesuai. Kemudian yang kedua, menggunakan *hardware* yaitu modul WARP sebagai implementasi tahap akhir sehingga dapat dilakukan pengambilan data secara *realtime* untuk dilakukan analisa. Sistem dirancang mulai dari proses pembangkitan bit hingga bit informasi diterima serta dilakukan deteksi sekaligus koreksi kesalahan menggunakan Kode Hamming. Secara umum sistem yang dirancang digambarkan pada blok diagram 3.1.



**Gambar 3.1** Blok Diagram Sistem Komunikasi M-QAM menggunakan Kode Hamming

Secara umum sistem terbagi menjadi dua, yaitu pemancar dan penerima. Setiap tahapan dari sistem yang telah dirancang akan dijelaskan dalam uraian berikut:

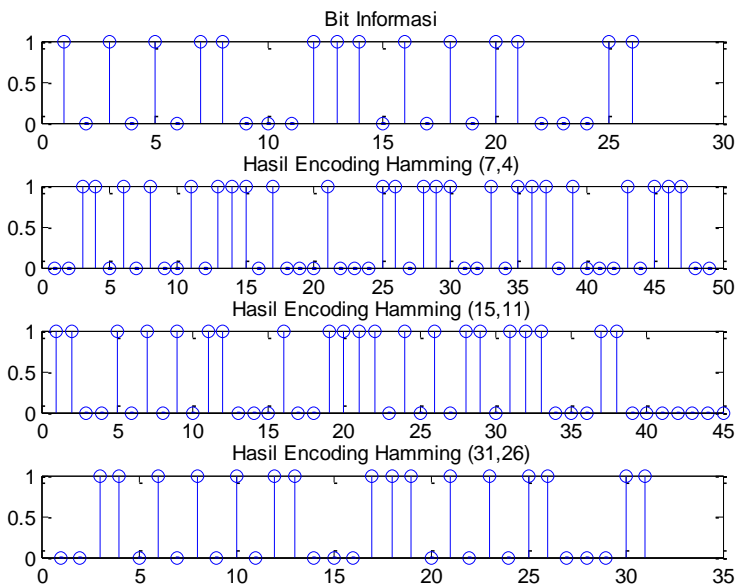
A. Blok Pemancar

1. Pembangkitan bit informasi

Pembangkitan bit informasi dilakukan secara acak dengan jumlah deretan bit disesuaikan. Namun pada saat pengukuran tetap menggunakan urutan bit acak yang sama, agar hasil pengukuran yang didapat mudah dibandingkan.

2. Encoding

Pada proses *encoding* deretan bit informasi dipisahkan per-blok untuk disisipkan bit *parity check* dari Kode Hamming. Besarnya blok informasi yang dipisahkan bergantung pada ukuran *code rate* yang digunakan. Hasil pembangkitan bit dan *encoding* dari 3 ukuran *code rate* Hamming ditunjukkan pada gambar 3.2.



**Gambar 3. 2** Hasil Pembangkitan Bit dan Encoding

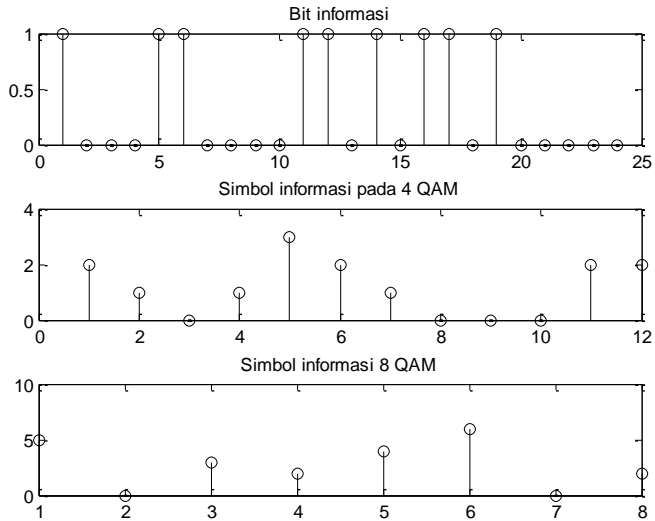
Pada proses *encoding* dilakukan penyisipan bit *parity check* yang diperoleh dengan cara mengalikan *codeword* (blok informasi) dengan generator matriks dari kode Hamming.



Jumlah bit *parity check* yang disisipkan tergantung pada ukuran *code rate* yang digunakan. Sehingga berpengaruh pada bandwidth yang digunakan. Karena semakin banyak bit *parity check* yang disisipkan maka kapasitas informasi yang dapat dikirimkan semakin berkurang. Begitu pula kualitas deteksi dan koreksi kesalahan yang dihasilkan juga berbeda.

3. Modulasi M-QAM

Bit-bit hasil *encoding* kemudian dibentuk deretan simbol-simbol berdasarkan level modulasi QAM. Perubahan bit informasi ke simbol dapat berpengaruh terhadap kapasitas bandwidth yang tersedia. Semakin tinggi level modulasi maka tingkat efisiensi bandwidth semakin meningkat. Hasil konversi bit menjadi simbol informasi untuk modulasi 4-QAM dan 8-QAM ditunjukkan pada gambar 3.3.



**Gambar 3.3** Hasil Konversi Bit menjadi Simbol Informasi

Simbol integer sebagai input dari proses modulasi, akan diubah menjadi bilangan kompleks sesuai dengan prinsip modulasi QAM yang memetakan setiap simbol informasi dalam konstelasi sinyal yang terbagi rata dalam 4 kuadran berdasarkan fase dan amplitudo. Hasil proses modulasi berupa simbol kompleks yang memiliki nilai real dan imajiner.

4. Penambahan Pilot

Selanjutnya, disela-sela bit informasi disisipkan sebuah simbol pilot (*estimator*) secara periodik yang berfungsi untuk proses estimasi kanal yang dilakukan di penerima. Hal ini diperlukan karena efek perubahan sinyal terkirim yang dihasilkan oleh kanal, perlu dilakukan estimasi agar pendeteksian sinyal informasi menjadi lebih akurat.

5. Penambahan *Preamble*

Selain penambahan pilot, deretan simbol tersebut ditambahkan *preamble* untuk menentukan letak simbol informasi pertama yang diterima di penerima. Begitu juga dengan *preamble*, nilai *preamble* sebelumnya telah diketahui oleh penerima.

6. *Upsample*

Selanjutnya dilakukan proses *Upsample* yang bertujuan untuk meningkatkan *sampling rate* dari data yang dikirim. Proses ini dilakukan dengan melewati ke filter *Squared Root Raised Cosine* (SRRC). Respon dari filter *raised cosine* adalah:

$$h(t) = \text{sinc}\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\beta t}{T}\right)}{1 - \left(\frac{2\beta t}{T}\right)^2} \quad (3.1)$$

dengan nilai  $\alpha$  adalah nilai *roll-off factor*;  $0 \leq \beta < 1$ .

7. *Upconvert*

Deretan informasi di-*upconvert* untuk mengkonversi sinyal ke frekuensi yang lebih tinggi dengan menggeser frekuensi tengah sinyal *baseband* ke frekuensi yang lebih tinggi. Pada pengerjaan tugas akhir ini bandwidth kanal yang digunakan 5 MHz dengan frekuensi kerja Wifi 2.4 GHz. Selain itu ada beberapa parameter lain yang digunakan dalam perancangan sistem akan dijelaskan pada tabel 3.1:

Proses *upconvert* bertujuan untuk menghindari pelemahan dari radio DC. *Upconvert* dilakukan dengan mengalikan sinyal terhadap frekuensi kompleks, yaitu:

$$T_{up} = s(t)xe^{j2\pi f_c t} \quad (3.2)$$

dengan  $f_c$  adalah frekuensi sampling,  $t$  adalah waktu.

**Tabel 3. 1** Parameter Sistem Komunikasi SISO pada WARP

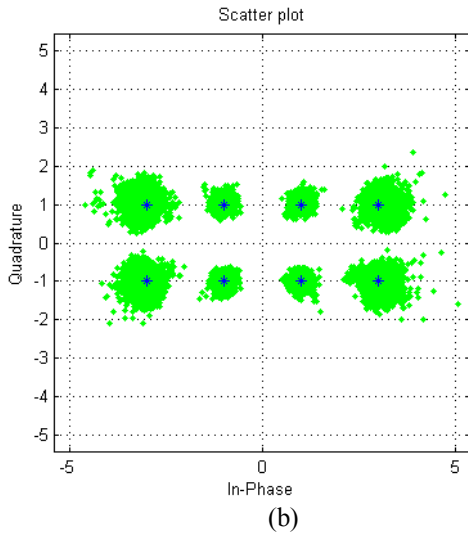
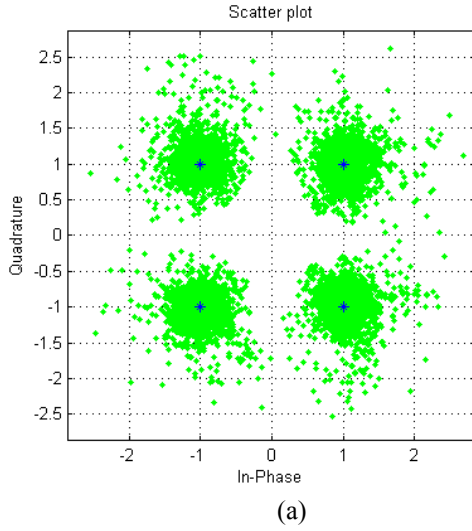
No	Parameter	Nilai
1	<i>Filter</i>	<i>Squared Root Raised Cosine</i>
2	<i>Filter Order</i>	64
3	<i>Roll Factor</i>	0.9
4	Jumlah <i>Preamble</i>	19
5	Jumlah corr window	300
6	Jumlah delay	400
7	Frekuensi <i>Upconvert</i>	5 MHz
8	Frekuensi Sampling	40 MHz
9	<i>Carrier channel</i>	Disesuaikan
10	<i>Transmitter baseband gain</i>	1
11	<i>Transmitter RF gain</i>	0-35 dB
12	<i>Receiver baseband gain</i>	1
13	<i>Receiver RF gain</i>	1

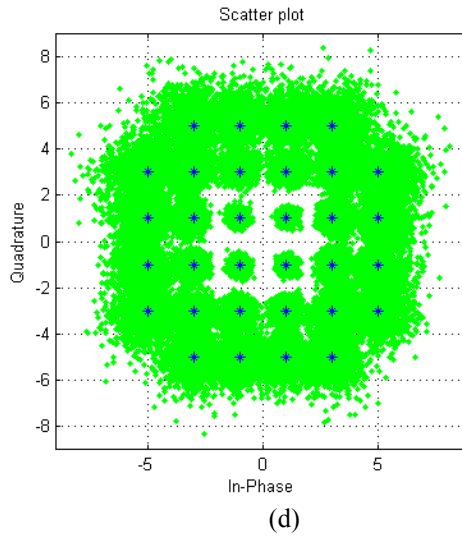
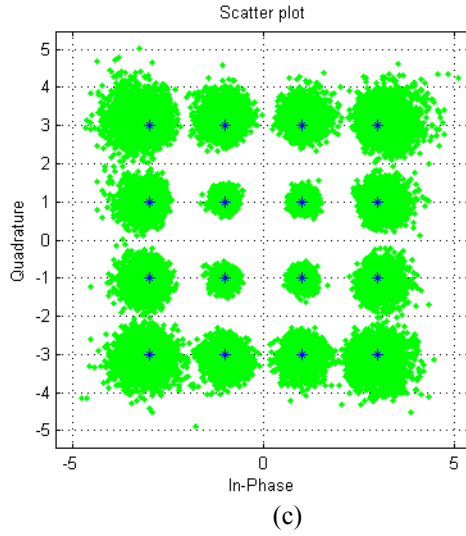
#### 8. *Transmit Data*

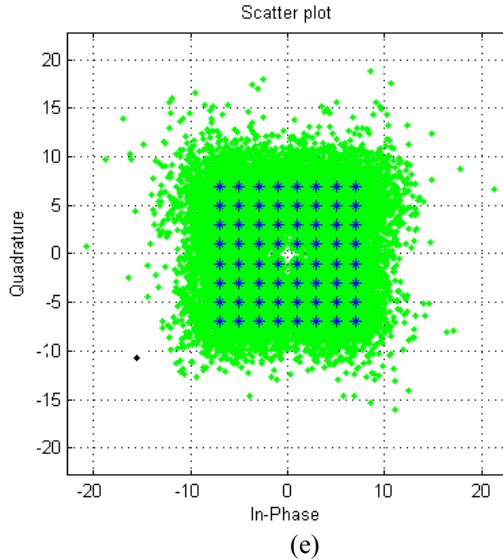
Sebelum proses pengiriman melalui WARP, sinyal yang akan dikirim dinormalisasi untuk mengoptimalkan DAC yang terdapat pada perangkat WARP. Pengoptimalan DAC pada nilai dengan range -1 sampai +1. Proses normalisasi dilakukan dengan cara menentukan nilai skala, kemudian dikalikan dengan sinyal hasil *upconvert*.

$$skala = \frac{1}{\text{nilai maks real dan imajiner sinyal hasil upconvert}} \quad (3.3)$$

Langkah selanjutnya, sinyal akan dikirimkan melalui radio pemancar pada perangkat WARP. Berikut *scatter plot* dari bit informasi yang dikirim dan terima berdasarkan level modulasi QAM ditunjukkan pada gambar 3.4.





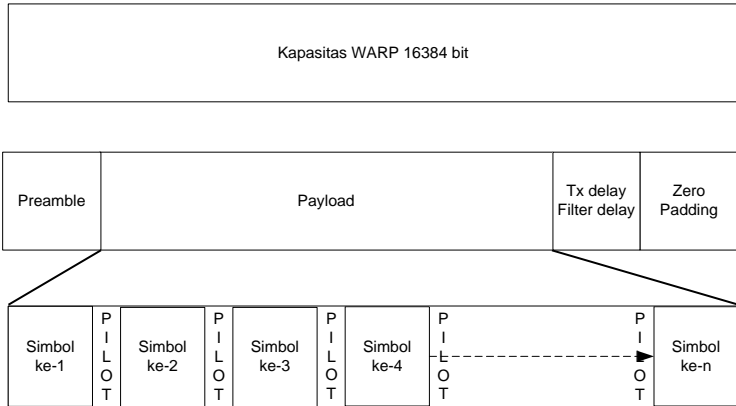


**Gambar 3. 4** Scater Plot Bit informasi yang Dikirim dan Diterima

(a) 4-QAM (b) 8-QAM (c) 16-QAM(d) 32-QAM (e) 64-QAM

Berdasarkan *scatter plot* diatas terlihat bahwa informasi yang dikirim dan diterima terbagi secara quadrature sesuai dengan level modulasi. Sebagian informasi yang diterima konstelasinya tidak sama dengan informasi aslinya. Hal ini disebabkan karena adanya noise yang timbul selama proses transmisi berlangsung sehingga menyebabkan bit error.

Setiap pengiriman data pada WARP dilakukan per-frame, dengan ukuran maksimum setiap frame  $2^{14}$  (16384 bit). Didalam frame terdiri dari bit informasi, *preamble*, *pilot*, *Tx delay*, *filter delay* dan *zero padding*. *Tx delay* berfungsi untuk mengantisipasi terjadinya delay pada saat transmisi agar data bisa diterima seluruhnya di penerima. Sedangkan *zero padding* ialah bit bernilai 0 yang ditambahkan apabila data yang ditransmisikan jumlahnya kurang dari kapasitas frame dari modul WARP. Berikut desain frame dari sistem komunikasi yang telah dirancang:



**Gambar 3.5** Desain Frame

**B. Blok Penerima**

1. *Received Data*

Pada penerima, sinyal yang diterima kembali oleh radio penerima pada perangkat WARP. Selanjutnya akan mengalami beberapa tahap yang berkebalikan dari beberapa proses yang ada di pengirim.

2. *Downconvert*

Sinyal di-downconvert dengan mengalikan sinyal konjugat dari frekuensi upconvert yaitu:

$$R_{down}(t) = r(t)xe^{-j2nfc t} \quad (3.4)$$

dengan  $fc$  adalah frekuensi sampling,  $t$  adalah waktu.

Spesifikasi frekuensi *sampling* pada WARP adalah 40 MHz.

3. *Matched Filtering*

Frekuensi tinggi yang dihasilkan dari proses *downconvert* akan dihilangkan dengan *matched filtering* agar sesuai dengan bentuk pulsa *raised cosine* di sisi pengirim.

4. *Pendeteksian Preamble*

Setelah proses *matched filtering*, dilakukan pendeteksian simbol pertama menggunakan *preamble*. Proses deteksi simbol pertama dilakukan berdasarkan korelasi tertinggi antara *sample* sinyal yang diterima dengan *sample preamble*.

5. *Downsample*

Selanjutnya sinyal di-*downsample* dengan jumlah sample saat *upsample* sebelumnya.

6. Pendeteksian pilot dan Estimasi kanal

Pendeteksian simbol yang dikirim dilakukan dengan estimasi kanal yang dilakukan dengan metode pilot. Proses yang terjadi pada estimasi kanal adalah dengan membandingkan sinyal pilot yang diterima dengan sinyal pilot referensi pada bagian penerima, sehingga respon kanal dari simbol tersebut dapat diperkirakan. Pada sisi penerima, simbol yang diterima dikalikan dengan nilai skala yang sama di sisi pengirim. Nilai simbol diketahui sebagai hasil kali dengan estimator karena asumsi tipe kanal yang *flat fading*.

$$\begin{aligned}y &= s \times h \\h^* \times y &= s \times h \times h^* \\h^* \times y &= s \times |h|^2 \\ \hat{s} &= \frac{y \times h^*}{|h|^2}\end{aligned}\tag{3.5}$$

Perbaikan fase dan amplitudo dapat dilakukan simbol estimasi ( $\hat{s}$ ), yang didapatkan dari hasil perkalian simbol diterima ( $y$ ) dengan konjugat estimasi kanal ( $h^*$ ) kemudian dibagi dengan kuadrat nilai mutlak dari estimasi kanal ( $|h|^2$ ).

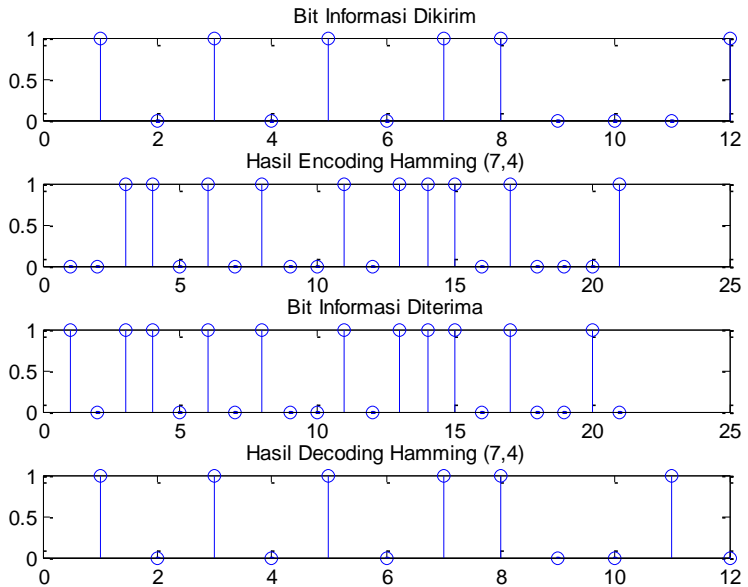
7. Demodulasi M-QAM

Deretan simbol selanjutnya didemodulasi untuk mendapatkan nilai simbol dalam bentuk integer.

8. *Decoding*

Kemudian deretan simbol yang dihasilkan diubah ke dalam bentuk bit dan dilakukan pendeteksian bit yang salah atau biasa disebut *decoding*. Untuk mengetahui *error* dalam *codeword*, dilakukan dengan membandingkan *syndrome error* dari masing-masing kode hamming dengan *syndrome error codeword* yang diterima. *Syndrome error* diperoleh dengan mengalikan *codeword* yang diterima dengan transpose matriks cek *parity*. Nilai *syndrome error* menentukan ada tidaknya *error* pada *codeword* yang diterima. Bila *syndrome error* bernilai nol, maka data yang diterima tidak terkena *error*. Berikut hasil decoding Hamming (7,4) ditunjukkan pada Gambar 3.6.





**Gambar 3. 6** Hasil *Decoding* Hamming (7,4)

Berdasarkan gambar 3.6, bit informasi yang dikirim berbeda dengan bit informasi yang diterima. Perbedaan tersebut berada pada bit pertama dan dua bit terakhir. Kemudian dilakukan deteksi dan koreksi melalui proses decoding, namun hanya bit pertama saja yang mampu diperbaiki, dan dua bit terakhir tetap salah. Kemampuan koreksi kode Hamming hanya satu bit per-bloknya. Kode Hamming mampu melakukan perbaikan error yang terjadi pada bit pertama. Dan error yang terjadi pada dua bit terakhir tidak mampu diperbaiki. Karena jumlah salah tersebut diluar batas kemampuan kerjanya.

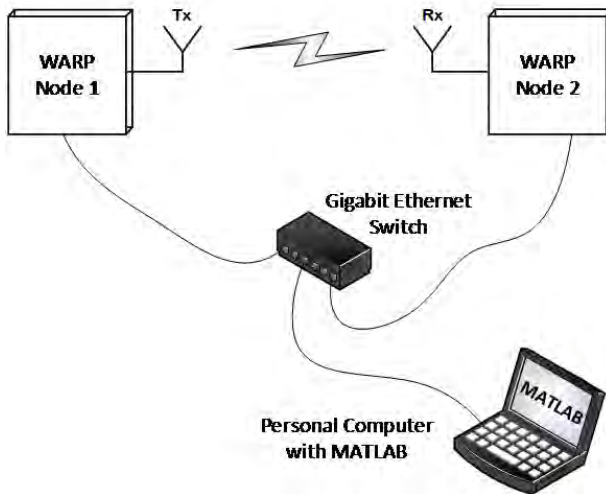
9. Bit diterima

Terakhir, bit informasi hasil decoding akan dihitung jumlah salahnya kemudian dibandingkan dengan jumlah bit yang dikirimkan sehingga dapat diketahui nilai BER dari hasil pengiriman sinyal informasi tersebut.

### 3.2 Implementasi pada Perangkat WARP

Sistem komunikasi yang dirancang disimulasikan menggunakan *software* Matlab untuk memastikan bahwa sistem telah sesuai dan berjalan dengan baik sebelum diimplementasikan pada WARP. Melalui WARP sinyal informasi yang dibangkitkan pada Matlab ditransmisikan secara *realtime*. Sehingga perlu dilakukan konfigurasi perangkat terlebih dahulu.

Pada sistem komunikasi SISO, membutuhkan 2 node WARP. Kedua WARP di integrasikan dengan PC melalui switch ethernet dan kabel LAN. Diagram konfigurasi perangkat WARP ditunjukkan pada gambar 3.9.



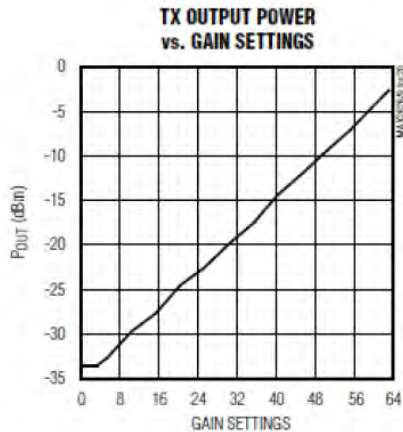
Gambar 3. 7 Konfigurasi modul WARP

Perangkat lunak dari WARP, WARPLab versi 7.1 akan diintegrasikan dengan PC. WARPLab digunakan agar sinyal yang dibangkitkan melalui *software* Matlab dapat dikirimkan oleh modul WARP. Sinyal informasi yang akan dikirim disimpan dalam *buffer Tx*. Setelah proses penyimpanan sinyal selesai, dilakukan persiapan proses transmisi dengan mengaktifkan radio pada node pengirim dan penerima.

Pada kedua modul WARP tersebut, terdapat *radio board* yang masing-masing berfungsi sebagai pemancar dan penerima

dengan power supply 5 Volt DC. Pengaturan gain pada *radio board* dilakukan melalui WARPLab dengan uraian sebagai berikut:

- a. *Transmitter Radio Frequency Gain*  
Perbandingan antara daya tx output dengan gain gain setting diperlihatkan pada Gambar 3.4
- b. *Transmitter Baseband Gain*  
Untuk baseband gain, terdapat 4 level pengaturan melalui WARPLab yaitu [1, 2, 3, 4] dengan nilai gain baseband dalam dB yaitu [-5, -3, -1.5, 0] dB [1]
- c. *Receiver Baseband Gain*  
Pada gain baseband receiver, terdapat 32 level pengaturan. Range pengaturan baseband receiver yaitu [0:31] dengan nilai dalam Db YAITU [0:63] dB[1]
- d. *Receiver Radio Frequency Gain*  
Pada pengaturan RF gain di penerima terdapat 3 level yaitu [0, 1, 2] dengan nilai dalam db yaitu [0, 15, 30] dB [1]



**Gambar 3. 8** Tx Output Power vs Gain Setting pada WARP[1]

Ketika radio board bekerja sebagai pemancar, sinyal digital akan diubah menjadi analog untuk dipancarkan. Begitu pula sebaliknya, sinyal yang dikirim akan diterima oleh node Rx. Proses pengambilan data dilakukan dengan pembacaan data yang terdapat pada *buffer Rx*.

### 3.3 Metode Pengukuran

Pada bagian ini akan diuraikan metode pengukuran yang digunakan. Secara keseluruhan pengukuran dilakukan secara *indoor* dan Line Of Sight (LOS) dengan menggunakan variasi daya pancar.



**Gambar 3. 9** Pengukuran menggunakan modul WARP

Pengukuran dilakukan di ruang 304 Laboratorium Teknik Telekomunikasi dan Multimedia jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Parameter pengukuran yang dilakukan yaitu:

#### 3.3.1 Level Modulasi

Level modulasi pada M-QAM merupakan parameter penting yang dijadikan acuan untuk mengetahui kinerja dari masing-masing nilai  $M$ -ary. Dalam pengukuran ini level modulasi yang digunakan sebanyak 5 level, yaitu 4, 8, 16, 32, 64.

### 3.3.2 Ukuran code rate Hamming

Kode Hamming untuk pengkodean kanal memiliki beberapa ukuran *code rate*. Ukuran *code rate* bergantung pada jumlah bit pariti yang disisipkan pada proses *encoding*. Pada pengukuran ini ukuran *code rate* yang digunakan sebanyak 3, yaitu (7,4); (15,11) dan (31,26).

### 3.3.3 Jarak Transmitter dan Receiver

Variasi jarak pemancar dan penerima dalam sistem komunikasi mempunyai pengaruh besar. Sehingga perlu dilakukan analisa mengenai pengaruh perubahan jarak terhadap kualitas informasi yang diterima. Pada pengukuran ini variasi jarak yang digunakan sebanyak 2, yaitu jarak 3 meter dan 6 meter.

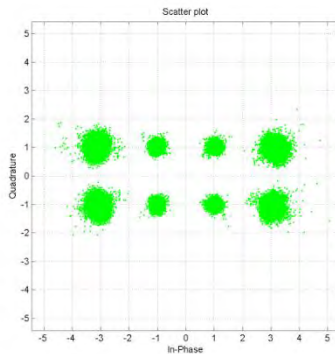
### 3.3.4 Perhitungan Eb/No

Pada implementasi menggunakan modul WARP, pengukuran dilakukan dengan memvariasikan daya pancar. Sehingga dapat diketahui nilai BER terhadap daya pancar. Namun untuk analisa lebih lanjut, maka hasil pengukuran perlu direpresentasikan dalam bentuk grafik BER terhadap Eb/No. Nilai Eb/No merupakan perbandingan antara besar energi bit dengan energi noise level.

Nilai Eb/No, diperoleh dari nilai real dan imajiner dari konstelasi sinyal yang diterima sebelum melalui proses demodulasi. Secara matematis, perhitungan Eb/No diuraikan sebagai berikut:

#### 1. Menghitung Eb

Nilai energi bit berasal dari nilai rata-rata energi simbol ( $E_s$ ) konstelasi yang diterima. Seperti yang ditunjukkan pada gambar 3.10.



**Gambar 3. 10** Konstelasi sinyal 8-QAM

Berdasarkan konstelasi tersebut, dapat dihitung energi masing-masing simbol:

$$E_{sM} = a^2 + b^2 \quad (3.6)$$

Dimana,  $a$  : level amplitudo *inphase*  
 $b$  : level amplitudo *quadrature*  
 $M$ : level modulasi

Energi simbol rata-rata:

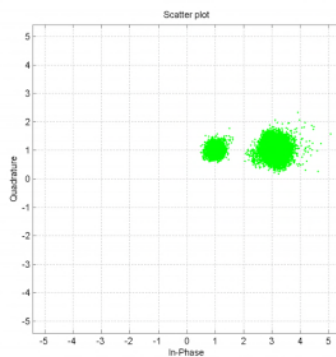
$$E_s = \frac{E_{s1} + E_{s2} + \dots + E_{sM}}{M} \quad (3.7)$$

Energi bit ( $E_b$ ):

$$E_b = \frac{E_s}{\log_2 M} \quad (3.8)$$

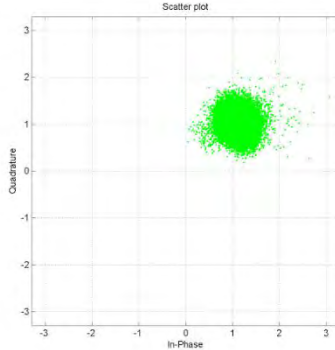
## 2. Menghitung $N_0$

Untuk menghitung variansi *noise*, maka semua titik konstelasi digeser ke daerah positif dengan memutlakan nilai amplitudo dari setiap simbol, seperti ditunjukkan pada gambar 3.11.



**Gambar 3. 11** Konstelasi sinyal 8-QAM digeser ke sisi positif

Kemudian titik konstelasi tersebut digabung menjadi satu titik tengah yang sama dengan menempatkan simbol pada posisi amplitudo *Inphase* dan *Quadrature* bernilai 1 (satu) seperti pada gambar 3.12.



**Gambar 3. 12** Penggabungan sinyal menjadi satu titik

Jadi untuk konstelasi simbol yang mempunyai amplitudo  $>1$ , maka harus dikurangi sedemikian rupa sehingga amplitudo bernilai 1.

$$\text{Inphase} = \text{abs}(\text{Nilai inphase})$$

$$\text{Quadrature} = \text{abs}(\text{Nilai Quadrature})$$

$$S = \text{inphase} + \text{quadrature} \quad (3.9)$$

Variasi noise

$$N_o = \frac{1}{N-1} \sum_{i=1}^N |S_i - \mu|^2 \quad (3.10)$$

Dimana  $\mu$  adalah rata – rata dari S.

3.  $E_b/N_o$

$$E_b/N_o(\text{dB}) = 10 \log_{10} \frac{E_b}{N_o} \quad (3.11)$$

*Halaman ini sengaja dikosongkan*

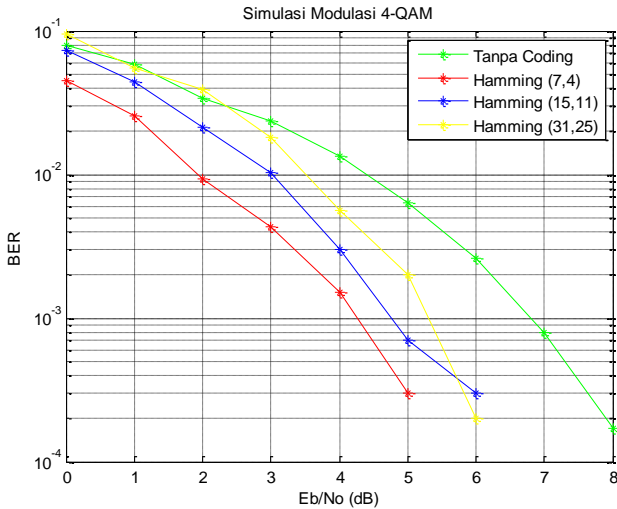


## BAB IV HASIL DAN ANALISA DATA

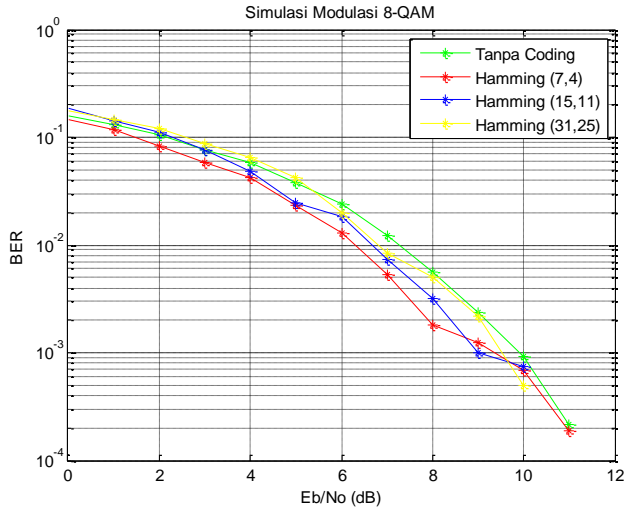
Pada bab ini akan dilakukan analisa mengenai kinerja pengkodean Hamming pada M-QAM berdasarkan hasil pengukuran dari sistem komunikasi yang telah diimplementasikan. Secara umum kinerja sistem dilihat dari nilai *Bit Error Rate* (BER) dengan menggunakan beberapa parameter yang sudah dijelaskan sebelumnya pada Bab 3. Uraian mengenai analisa dari hasil pengukuran dijabarkan sebagai berikut.

### 4.1 Modulasi M-QAM Dengan dan Tanpa Kode Hamming

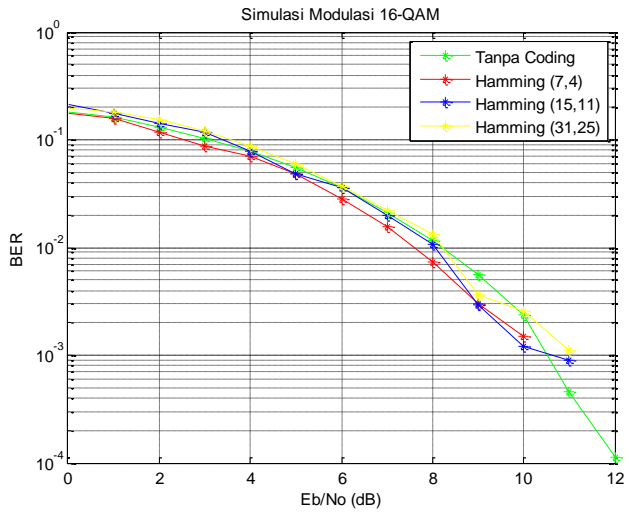
Sebelum melakukan implementasi sistem pada perangkat WARP, telah dilakukan proses simulasi menggunakan kanal AWGN. Hal ini bertujuan untuk memastikan bahwa sistem yang dirancang telah berjalan dengan baik dan sesuai teori. Sehingga dari hasil pengukuran dapat dianalisa dengan mengacu pada hasil simulasi. Berikut hasil simulasi dari penggunaan kode Hamming pada modulasi M-QAM ditunjukkan pada gambar 4.1



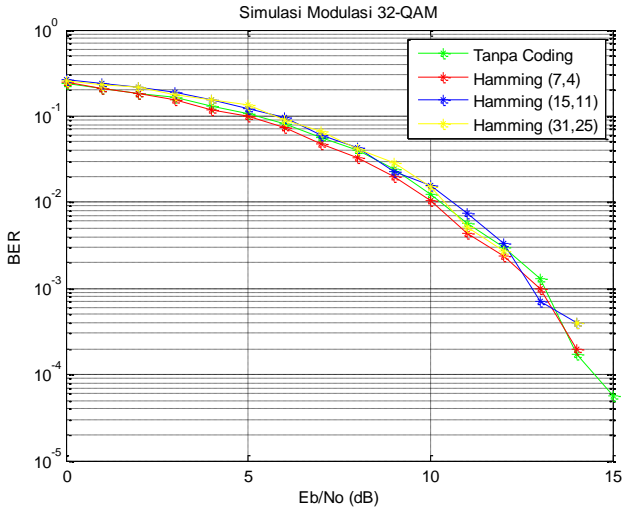
(a)



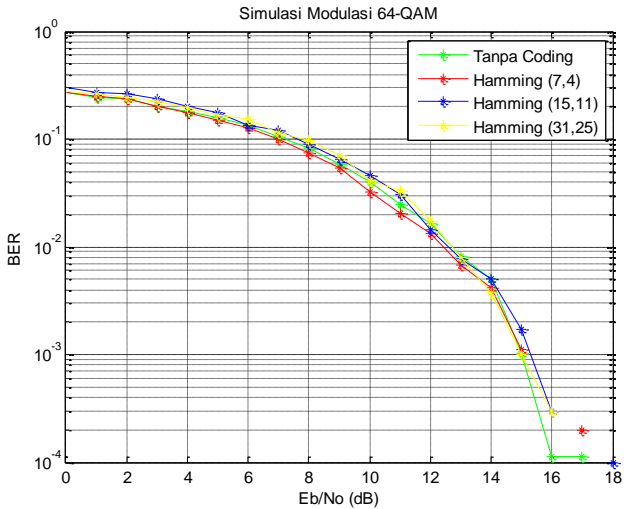
(b)



(c)



(d)



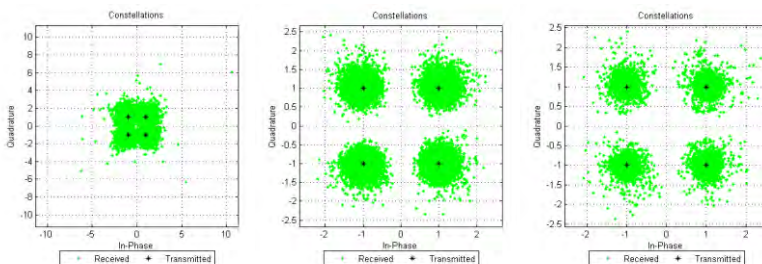
(e)

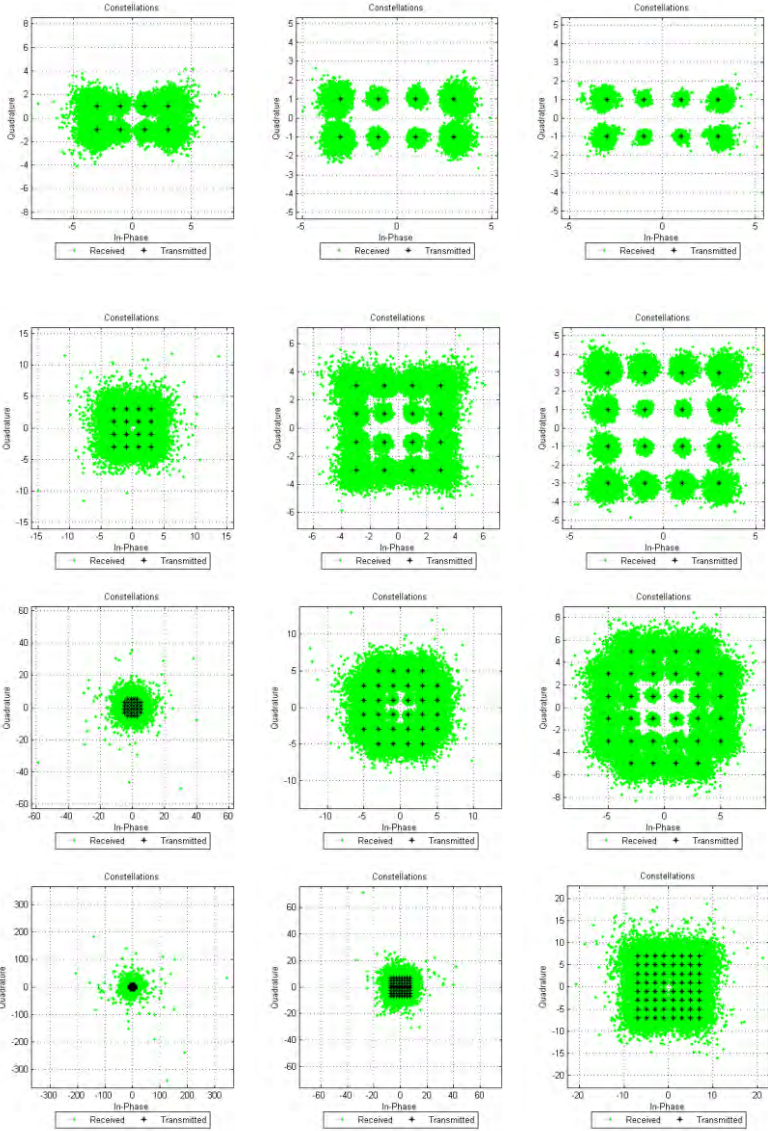
**Gambar 4. 1** Grafik BER Hasil Simulasi Modulasi M-QAM (a) 4-QAM; (b) 8-QAM (c); 16-QAM; (d) 32-QAM; (e)64-QAM

Berdasarkan hasil simulasi yang telah dilakukan, menunjukkan bahwa kinerja kode Hamming pada masing-masing level modulasi QAM berbeda. Hal ini disebabkan karena probabilitas error dari setiap level modulasi tidak sama. Pada dasarnya, kode Hamming hanya mampu melakukan perbaikan satu bit error setiap bloknya. Sehingga pada level modulasi yang mempunyai probabilitas error tinggi, maka kode Hamming tidak bisa bekerja secara maksimal. Seperti yang ditunjukkan pada gambar 4.1 poin (d) dan (e), perbaikan tidak seberapa terlihat. Berbeda halnya pada saat kode Hamming digunakan pada level modulasi rendah khususnya 4-QAM, perbaikan error terlihat jelas dari grafik BER yang mengalami penurunan karena probabilitas error pada level ini lebih kecil. Penurunan nilai BER yang paling tinggi yaitu dengan menggunakan kode Hamming (7,4) dilanjutkan dengan Hamming (15,11) kemudian (31,26). Sesuai pada teori, Hamming (7,4) memiliki kemampuan koreksi yang paling baik, karena diimbangi dengan banyaknya bit *parity* yang disisipkan. Sehingga meskipun (7,4) memiliki kinerja yang paling baik, namun efisiensi bandwidtnya kecil.

## 4.2 Konstelasi Simbol pada Penerima

Konstelasi sinyal pada M-QAM merepresentasikan jumlah simbol dalam setiap level modulasi. Kualitas konstelasi yang dihasilkan pada penerima salah satunya dipengaruhi oleh besarnya daya pancar yang digunakan. Karena pengiriman informasi dengan daya yang kecil akan menyebabkan sinyal yang diterima juga lemah sehingga probabilitas *error* yang terjadi semakin besar. Begitu pula sebaliknya, penggunaan daya pancar yang lebih besar mampu mengurangi terjadinya error bit sehingga konstelasi yang dihasilkan lebih baik, seperti ditunjukkan pada gambar 4.2.



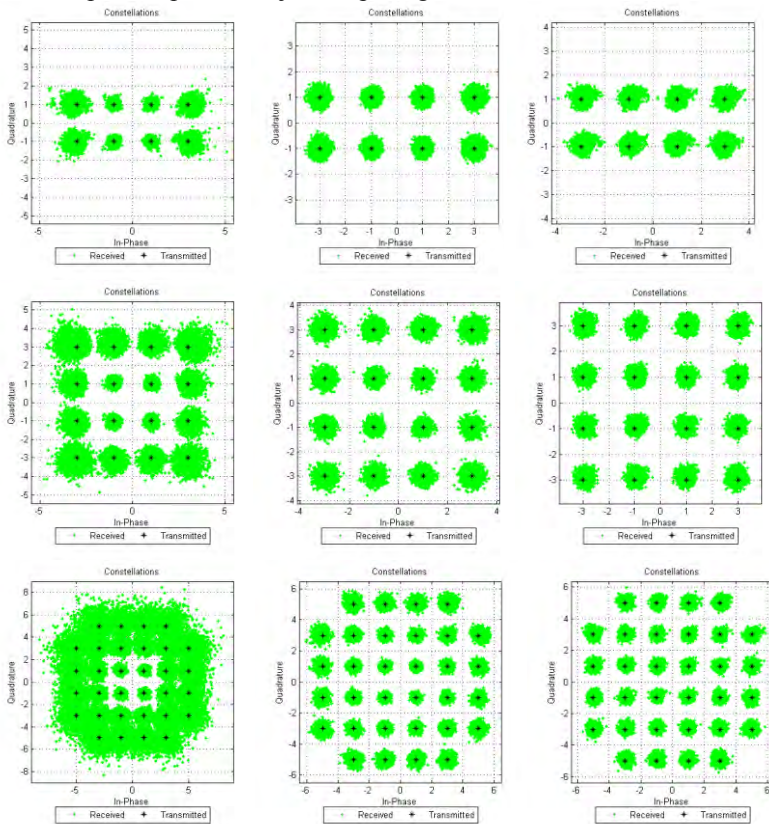


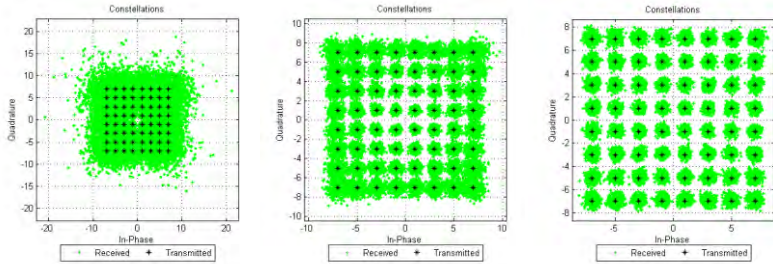
(a) Pt = -27 dBm      (b) Pt = -21,25 dBm      (c) Pt = -14,5 dBm

**Gambar 4. 2** Konstelasi Sinyal M-QAM (M-ary 4, 8, 16, 32, dan 64) dengan variasi daya pancar

Berdasarkan gambar diatas dapat dilihat pengaruh daya pancar terhadap konstelasi sinyal. Saat daya pancar kecil, sinyal yang diterima tampak lebih menyebar. Penyebaran titik konstelasi ini menyebabkan penerima lebih sulit untuk menentukan *decision area* (daerah keputusan) sehingga probabilitas *error* semakin besar.

Pada hasil pengukuran didapatkan bahwa penggunaan daya pancar yang tinggi tidak sepenuhnya mampu meminimalisir penyebaran sinyal khususnya pada level modulasi yang mempunyai nilai amplitudo sinyal lebih besar dari satu ( $M\text{-ary} > 4$ ). Maka, dilakukan perubahan nilai amplitudo pilot yang disisipkan, amplitudo pilot yang digunakan sebelumnya bernilai satu. Untuk hasil konstelasi dari beberapa variasi nilai amplitudo pilot ditunjukkan pada gambar 4.3.



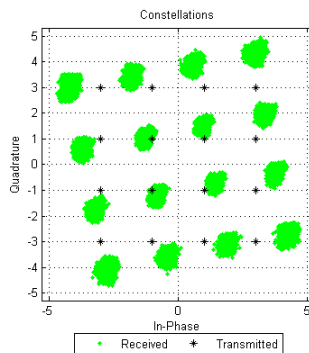


(a) amplitudo pilot 1      (b) amplitudo pilot 4      (c) amplitudo pilot 10

**Gambar 4. 3** Konstelasi sinyal dengan variasi nilai amplitudo pilot

Konstelasi yang ditunjukkan pada gambar 4.2 yaitu pada saat daya pancar 14,5 dBm. Pada gambar hanya ditampilkan konstelasi sinyal untuk level modulasi 8, 16, 32, dan 64. Hal ini dikarenakan penyebaran titik terluar hanya terjadi pada level modulasi yang mempunyai amplitudo sinyal lebih besar dari satu.

Berdasarkan gambar diatas dapat dilihat bahwa penambahan nilai amplitudo pada pilot dapat memberikan perubahan. Dimana penyebaran titik konstelasi pada sisi terluar dapat dikurangi, sehingga konstelasi yang dihasilkan menjadi lebih baik. Karena *error* bit yang dihasilkan juga berkurang. Namun, penambahan amplitudo pilot pada saat daya pancar tinggi menyebabkan terjadinya pergeseran fasa. Sehingga sinyal tampak berputar atau bergeser dari sebelumnya. Seperti yang ditunjukkan pada gambar 4.4. Untuk hasil konstelasi lebih lengkap ada pada lampiran C.



**Gambar 4. 4** Konstelasi yang mengalami pergeseran fasa

Pada gambar 4.3, daya pancar yang digunakan -7 dBm dengan amplitudo pilot 10. Pergeseran fasa tersebut disebabkan karena amplifier untuk perangkat WARP mempunyai kecenderungan penguatan yang tidak linier pada saat daya yang dipancarkan tinggi. Sehingga mengalami distorsi fasa yang mengakibatkan nilai *Error Vector Magnitude* (EVM) yang tinggi.

### 4.3 Perbandingan BER dalam Variasi Level Modulasi

Perubahan level modulasi pada teknik modulasi M-QAM memberikan pengaruh terhadap kualitas performansi. Level modulasi akan mempengaruhi kapasitas pengiriman dan kualitas bit informasi yang dikirimkan. Semakin tinggi level modulasi maka efisiensi bandwidth juga semakin tinggi. Namun tingkat kesalahan bit yang terjadi pada proses transmisi semakin besar. Hal ini dikarenakan level modulasi berbanding lurus dengan jumlah simbol yang terbentuk. Semakin banyak variasi simbol yang dihasilkan maka penerima akan lebih sulit dalam menentukan *decision area* (daerah keputusan) sehingga menyebabkan peluang terjadinya kesalahan semakin besar. Penggunaan kode Hamming pada teknik modulasi M-QAM mampu memberikan perbaikan terhadap kinerja sistem, nilai BER yang dihasilkan dapat ditekan lebih kecil. Hasil pengukuran dalam variasi level modulasi tanpa dan dengan menggunakan kode Hamming pada jarak 6 meter ditunjukkan pada tabel 4.1 sampai 4.4.

**Tabel 4. 1** Hasil Pengukuran Tanpa Kode Hamming Jarak 6 M

PTx (dBm)	4 QAM	8 QAM	16 QAM	32 QAM	64 QAM
-35	0,06037	0,10545	0,23904	0,38073	0,48859
-32.5	0,04183	0,08312	0,20743	0,34826	0,46765
-30	0,02619	0,03613	0,13683	0,27867	0,40753
-27.5	0,00988	0,02076	0,09388	0,23773	0,34989
-25	0,00159	0,00773	0,05109	0,17155	0,29037
-22.5	0,00024	0,00413	0,03257	0,13351	0,25509
-20	0,00001	0,00175	0,01532	0,08620	0,21951
-17.5	0,00000	0,00126	0,01050	0,06916	0,18432
-15	0,00000	0,00056	0,00508	0,04437	0,15083
-12.5	0,00000	0,00035	0,00363	0,04661	0,13179



**Tabel 4. 2** Hasil Pengukuran dengan Kode Hamming (7,4) Jarak 6 M

<b>PTx (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,05544	0,09452	0,23402	0,38131	0,48131
-32.5	0,03219	0,07024	0,20298	0,35198	0,46896
-30	0,01654	0,02688	0,12719	0,28393	0,40667
-27.5	0,00447	0,01401	0,08325	0,23581	0,35551
-25	0,00061	0,00438	0,03657	0,16848	0,29388
-22.5	0,00002	0,00247	0,02035	0,12806	0,25705
-20	0,00000	0,00122	0,00758	0,08341	0,20547
-17.5	0,00000	0,00063	0,00496	0,05866	0,18123
-15	0,00000	0,00025	0,00221	0,04333	0,13569
-12.5	0,00000	0,00013	0,00108	0,03325	0,12561

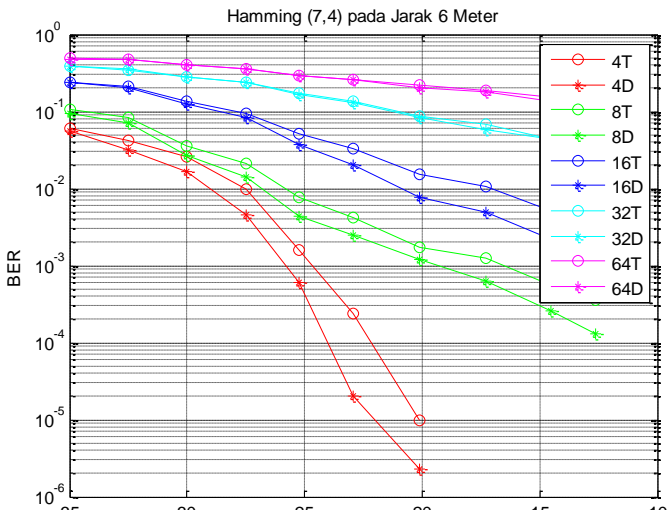
**Tabel 4. 3** Hasil Pengukuran dengan Kode Hamming (15,11) Jarak 6 M

<b>PTx (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,06253	0,11794	0,24477	0,38414	0,48137
-32.5	0,03418	0,09190	0,21034	0,35713	0,47782
-30	0,01822	0,03617	0,13070	0,29298	0,41390
-27.5	0,00692	0,01634	0,08795	0,24457	0,36233
-25	0,00097	0,00560	0,03634	0,17732	0,30514
-22.5	0,00006	0,00290	0,02021	0,14245	0,27234
-20	0,00001	0,00116	0,00734	0,08917	0,22534
-17.5	0,00000	0,00056	0,00350	0,06359	0,20235
-15	0,00000	0,00040	0,00216	0,03971	0,15549
-12.5	0,00000	0,00043	0,00185	0,03011	0,14288

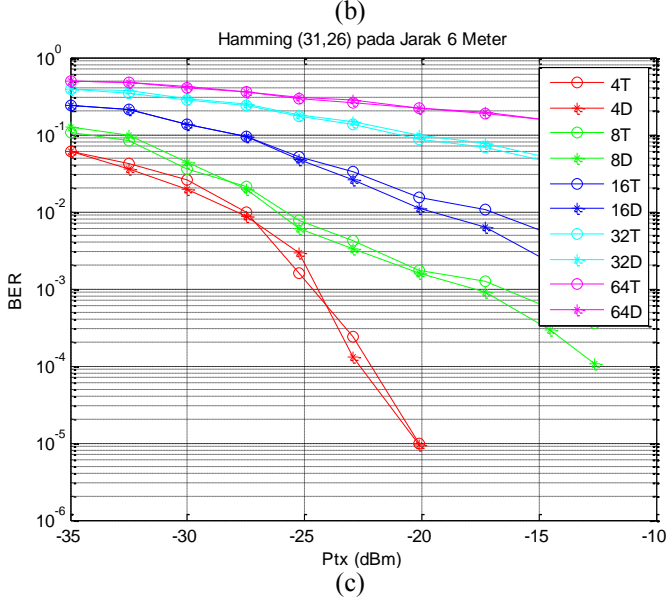
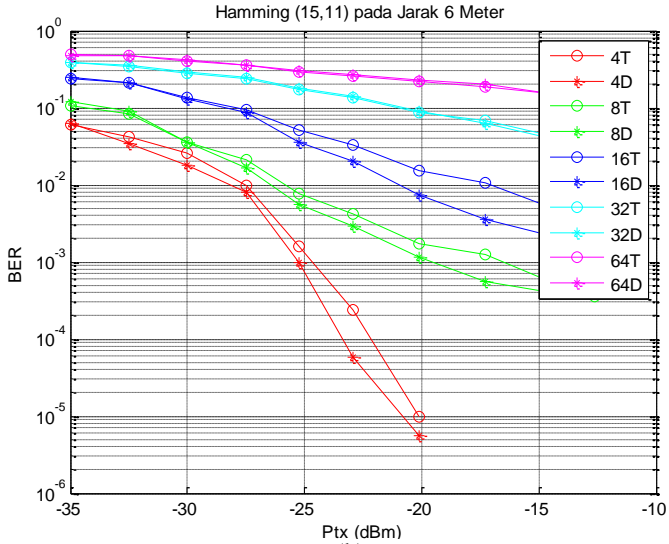
**Tabel 4. 4** Hasil Pengukuran dengan Kode Hamming (31,26) Jarak 6 M

<b>PT<sub>x</sub> (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,06036	0,12301	0,23995	0,38400	0,48544
-32.5	0,03641	0,09884	0,20873	0,36451	0,48138
-30	0,01927	0,04409	0,13797	0,29093	0,41816
-27.5	0,00753	0,01981	0,09591	0,24928	0,36266
-25	0,00286	0,00595	0,04726	0,18109	0,30510
-22.5	0,00013	0,00334	0,02605	0,14472	0,27935
-20	0,00001	0,00158	0,01116	0,09685	0,22407
-17.5	0,00000	0,00090	0,00626	0,07678	0,19556
-15	0,00000	0,00029	0,00222	0,04898	0,15401
-12.5	0,00000	0,00011	0,00163	0,03654	0,15109

Terlihat pada keempat tabel tersebut, menunjukkan bahwa nilai BER dari setiap modulasi mengalami peningkatan seiring bertambahnya level modulasi. Besarnya nilai BER berbanding lurus dengan level modulasi. Berikut Grafik perbandingan BER dalam variasi Level Modulasi ditunjukkan pada gambar 4.5.



(a)  
56



**Gambar 4.5** Grafik BER Terhadap Daya Pancar 1  
 (a)M-QAM menggunakan Kode Hamming (7,4)  
 (b)M-QAM menggunakan Kode Hamming (15,11)  
 (c)M-QAM menggunakan Kode Hamming (31,26)

Keterangan legenda pada gambar 4.5:

1. Angka menunjukkan level modulasi
2. Huruf T (Tanpa) dan D (dengan) menggunakan kode Hamming

Misal:

4T : Modulasi 4-QAM tanpa menggunakan kode Hamming

4D : Modulasi 4-QAM dengan menggunakan kode Hamming

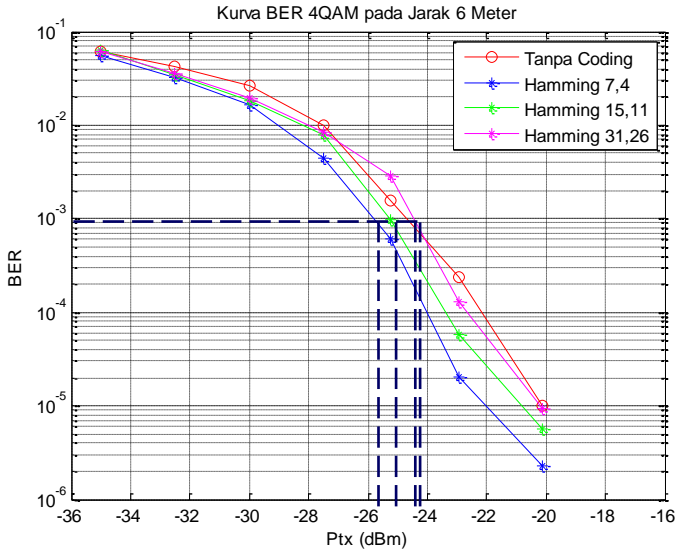
Penggunaan kode Hamming pada pada level modulasi 4, 8 dan 16 QAM mampu memberikan perbaikan yang terlihat jelas seperti pada grafik 4.5. Namun perbaikan kode Hamming pada level modulasi 32 dan 64 QAM tidak sebaik 3 level modulasi sebelumnya. Penggunaan pengkodean kanal pada level modulasi tersebut juga kurang terlihat memberikan perbaikan yang signifikan. Hal ini disebabkan karena tingginya level modulasi menyebabkan jumlah error semakin besar. Sehingga untuk kesalahan bit yang lebih dari satu dalam setiap bloknya, kode Hamming tidak mampu melakukan perbaikan. Karena kemampuan koreksi kode Hamming hanya satu bit setiap bloknya.

#### 4.4 Perbandingan BER dalam Variasi Ukuran Code Rate

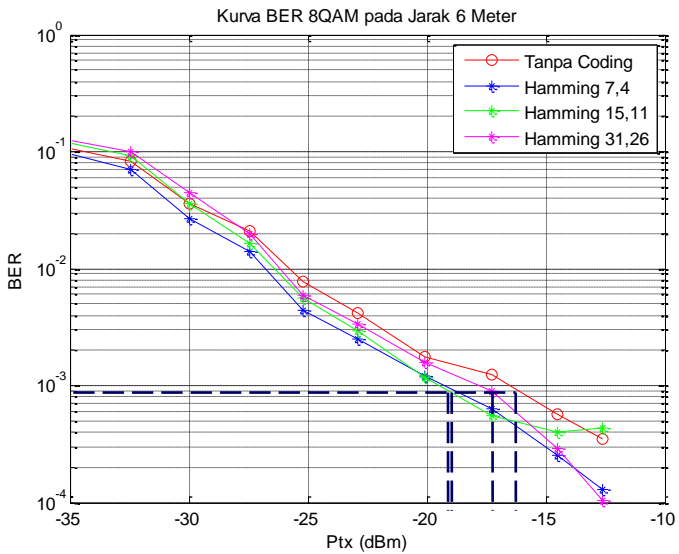
Variasi ukuran *code rate* yang digunakan pada penelitian ini sebanyak 3 macam, yaitu Hamming (7,4); Hamming (15,11) dan Hamming (31,26). Ketiga ukuran *code rate* tersebut dibedakan dengan jumlah bit *parity check* yang disisipkan. Selain berpengaruh terhadap kapasitas pengiriman bit informasi, bit *parity check* memberikan pengaruh terhadap kualitas perbaikan bit informasi yang diterima. Misalnya pada code rate Hamming (7,4), berarti jumlah bit informasi per-blok adalah 4 bit dan 3 bit lainnya adalah bit *parity check*. Begitu pula pada ukuran *code rate* yang lain. Blok informasi (*codeword*) yang panjang akan memberikan peluang kesalahan bit lebih besar dari satu semakin banyak. Seperti yang sudah dijelaskan pada bab 2, bahwa kode Hamming hanya mampu melakukan perbaikan untuk satu bit *error* per-blok. Dan jika kesalahan lebih dari satu, perbaikan yang dilakukan kode Hamming akan menghasilkan koreksi yang salah. Atau bahkan jumlah *error* lebih banyak dari sebelumnya.

Di lain sisi, minimnya bit *parity check* yang disisipkan pada setiap blok akan memberikan efisiensi bandwidth yang besar. Namun kualitas koreksinya menjadi menurun. Seiring dengan bertambah panjangnya blok maka peluang *error*-nya juga semakin besar. Mengacu pada tabel 4.1 sampai 4.4, pada subbab sebelumnya berikut grafik

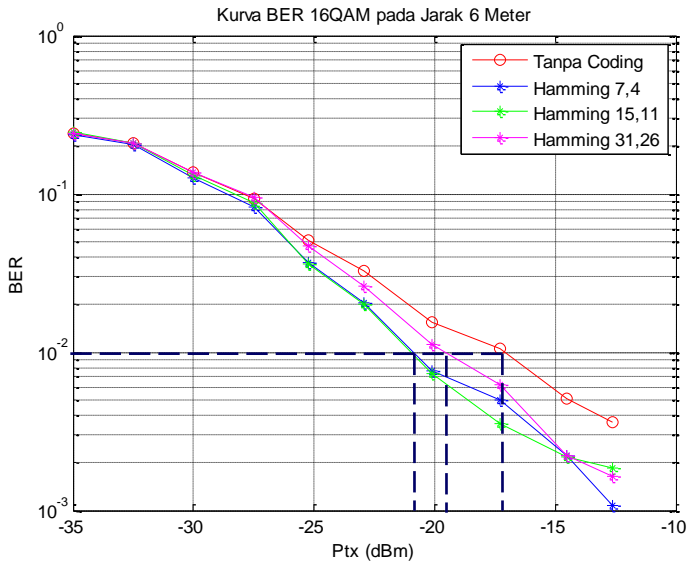
perbandingan BER dalam variasi ukuran *code rate* yang ditunjukkan pada Gambar 4.6.



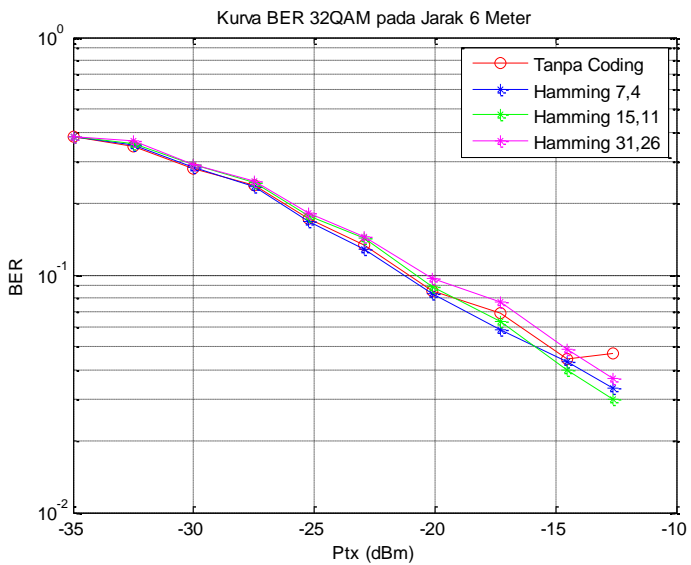
(a)



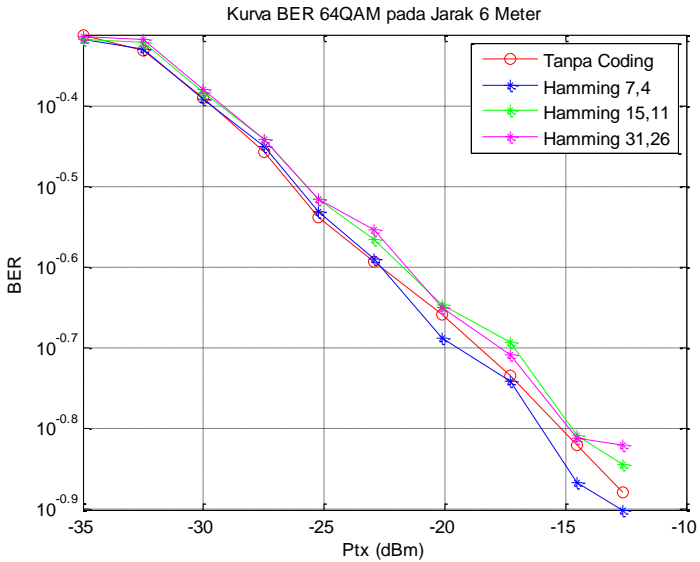
(b)



(c)



(d)



(e)

**Gambar 4. 6** Grafik BER Terhadap Daya Pancar 2  
 (a)4-QAM Tanpa dan Menggunakan Kode Hamming  
 (b)8-QAM Tanpa dan Menggunakan Kode Hamming  
 (c)16-QAM Tanpa dan Menggunakan Kode Hamming  
 (d)32-QAM Tanpa dan Menggunakan Kode Hamming  
 (e)64-QAM Tanpa dan Menggunakan Kode Hamming

Berikut gain code pada ketiga ukuran kode Hamming dari 3 level modulasi terkecil.

**Tabel 4. 5** Gain Code untuk Jarak 6 Meter

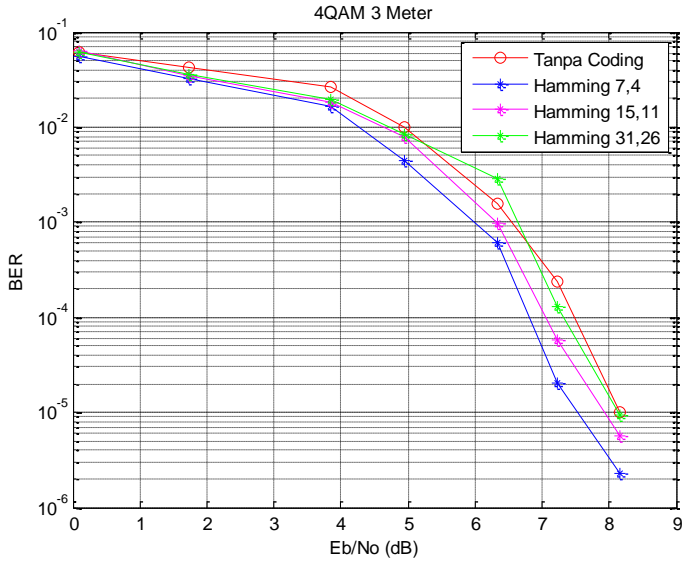
Modulasi	BER	Tanpa Kode	Dengan Kode Hamming		
			(7,4)	(15,11)	(31,26)
4-QAM	$10^{-3}$	-24.5	-25.9	-25.2	-24.4
8-QAM	$10^{-3}$	-17	-19.1	-19.2	-18
16-QAM	$10^{-2}$	-17.2	-24	-24	-19,1

Pada modulasi 64-QAM dan 32-QAM saat BER bernilai  $10^{-1}$ , perbedaan gain code pada setiap ukuran code rate sangat kecil. Ini menunjukkan bahwa perbedaan perbaikan antar code rate selisihnya juga kecil. Terlebih pada level modulasi tersebut penggunaan kode Hamming menjadikan nilai BER lebih tinggi daripada tanpa menggunakan pengkodean kanal. Dikarenakan keterbatasan kemampuan koreksi kode Hamming. Meskipun kode Hamming mampu mendeteksi hingga dua bit error setiap bloknya, namun tidak mampu melakukan koreksi jika hal tersebut terjadi. Sehingga apabila terjadi dua bit error dalam satu blok maka perbaikan yang dilakukan juga tetap salah.

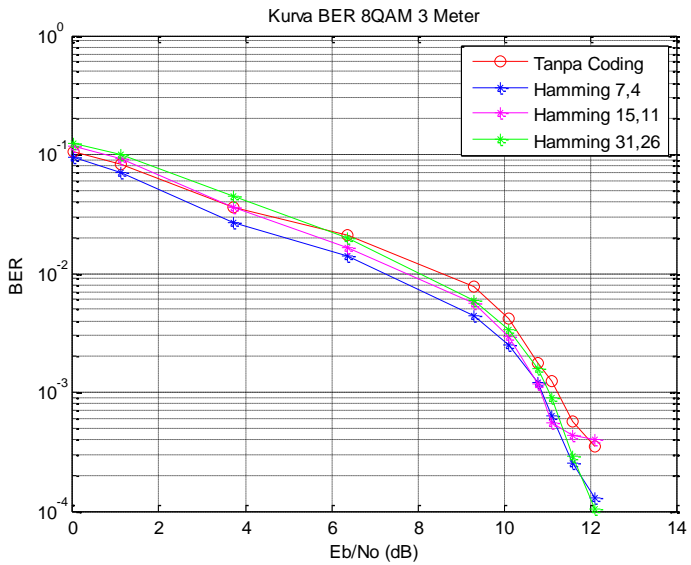
Untuk modulasi 16-QAM gain code dilihat pada saat BER  $10^{-2}$ . Karena variasi daya pancar terbatas sehingga tidak mencapai BER  $10^{-3}$ . Pada modulasi 8-QAM kode Hamming (7,4) dan (15,11) memerlukan daya pancar yang hampir sama untuk mencapai nilai BER  $10^{-3}$ . Selisih gain code untuk masing-masing code rate  $\leq 1$  dB. Pada modulasi 4-QAM menggunakan kode Hamming (7,4) perbaikan mencapai kurang lebih 1.4 dB, sedangkan untuk kode Hamming (15,11) perbaikan 0.6 dB dan untuk Hamming (31,26) perbaikan 0.1 dB. Sehingga dari ketiga code rate tersebut, code rate Hamming (7,4) mampu memberikan perbaikan yang lebih baik dibandingkan dengan ukuran code rate yang lain.

Ditinjau dari nilai BER terhadap Eb/No, kinerja kode Hamming menunjukkan hal yang tidak beda jauh dengan grafik BER terhadap daya pancar. Setiap ukuran *code rate* menunjukkan bahwa penggunaan kode Hamming mampu memberikan perbaikan daripada tanpa menggunakan kode Hamming dengan tingkat perbaikan yang berbeda-beda. Hanya saja perbaikan ini terbatas oleh level modulasi dari QAM. Semakin tinggi level modulasi perbaikan kode Hamming tidak terlihat secara signifikan. Hal ini disebabkan karena semakin tinggi level modulasi maka tingkat probabilitas error semakin meningkat. Selain itu disebabkan karena kemampuan kode Hamming yang hanya mampu mengoreksi satu bit error dalam setiap bloknya. Grafik BER terhadap Eb/No ditunjukkan pada Gambar 4.7.

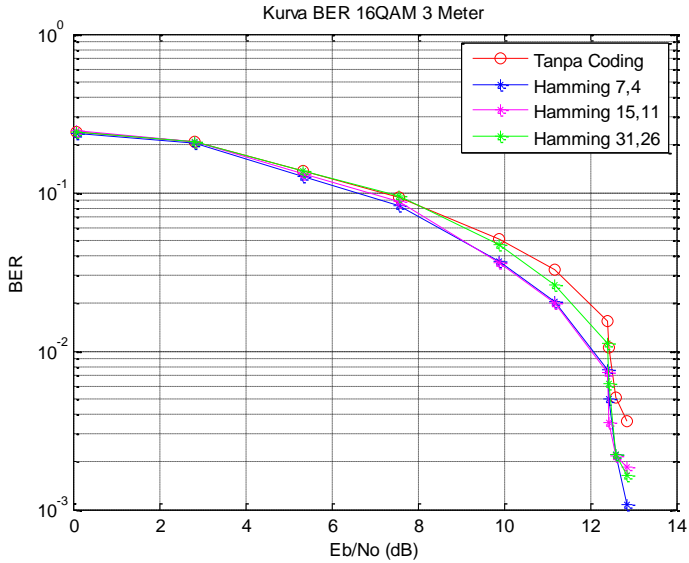




(a)



(b)



(c)

**Gambar 4. 7** Grafik BER Terhadap  $E_b/N_0$

- (a)4-QAM Tanpa dan Menggunakan Kode Hamming
- (b)8-QAM Tanpa dan Menggunakan Kode Hamming
- (c)16-QAM Tanpa dan Menggunakan Kode Hamming

#### 4.5 Perbandingan BER dalam Variasi Jarak

Jarak pemancar dan penerima akan memberikan pengaruh terhadap kualitas informasi pada sistem komunikasi. Semakin jauh jarak maka nilai BER akan semakin besar pula. Hal ini disebabkan karena jarak yang lebih jauh akan memberikan peluang terjadinya gangguan atau noise semakin besar. Dan tidak menutup kemungkinan juga jarak dekat akan memberikan nilai BER yang kurang bagus dibanding dengan jarak yang jauh. Karena juga tergantung besarnya noise, ataupun penghalang transmisi disekitar daerah pengukuran. Sehingga untuk mengantisipasi hal tersebut maka pengukuran dengan variasi jarak dilakukan dalam kondisi ruang yang sama. Berikut hasil pengukuran pada jarak 3 meter ditunjukkan pada tabel 4.6 sampai 4.9.

**Tabel 4. 6** Hasil Pengukuran Tanpa Kode Hamming Jarak 3 M

<b>TANPA</b>					
<b>PT<sub>x</sub> (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,03725	0,05695	0,18959	0,33928	0,44923
-32.5	0,02599	0,04178	0,15658	0,30089	0,43256
-30	0,00522	0,01298	0,09643	0,24119	0,35440
-27.5	0,00095	0,00437	0,05420	0,19147	0,30957
-25	0,00001	0,00137	0,01778	0,15236	0,26110
-22.5	0,00000	0,00014	0,00813	0,09901	0,23434
-20	0,00000	0,00004	0,00373	0,05349	0,17988
-17.5	0,00000	0,00002	0,00082	0,03428	0,15232
-15	0,00000	0,00001	0,00024	0,02031	0,12895
-12.5	0,00000	0,00000	0,00001	0,01668	0,11187

**Tabel 4. 7** Hasil Pengukuran dengan Kode Hamming(7,4) Jarak 3 M

<b>PT<sub>x</sub> (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,03202	0,04656	0,16141	0,32854	0,45118
-32.5	0,01703	0,02986	0,13262	0,30383	0,42545
-30	0,00103	0,00957	0,07361	0,23461	0,34675
-27.5	0,00005	0,00247	0,03932	0,17589	0,30997
-25	0,00000	0,00026	0,01283	0,13336	0,25249
-22.5	0,00000	0,00003	0,00434	0,07562	0,22330
-20	0,00000	0,00002	0,00094	0,04258	0,17589
-17.5	0,00000	0,00001	0,00017	0,02749	0,13622
-15	0,00000	0,00000	0,00001	0,01070	0,11062
-12.5	0,00000	0,00000	0,00000	0,00917	0,09644

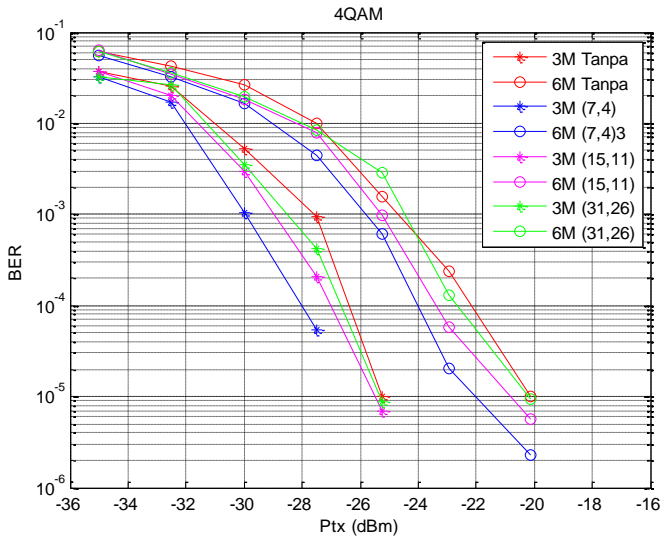
**Tabel 4. 8** Hasil Pengukuran dengan Kode Hamming(15,11) Jarak 3 M

<b>PT<sub>x</sub> (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,03670	0,06795	0,17510	0,33533	0,46570
-32.5	0,02013	0,04923	0,14235	0,31455	0,45303
-30	0,00294	0,01308	0,08206	0,24353	0,37092
-27.5	0,00021	0,00479	0,04736	0,20208	0,32114
-25	0,00001	0,00054	0,01315	0,14102	0,26908
-22.5	0,00000	0,00005	0,00660	0,08079	0,23482
-20	0,00000	0,00002	0,00189	0,04597	0,18274
-17.5	0,00000	0,00001	0,00028	0,02965	0,16113
-15	0,00000	0,00001	0,00001	0,01589	0,12186
-12.5	0,00000	0,00000	0,00000	0,01120	0,11500

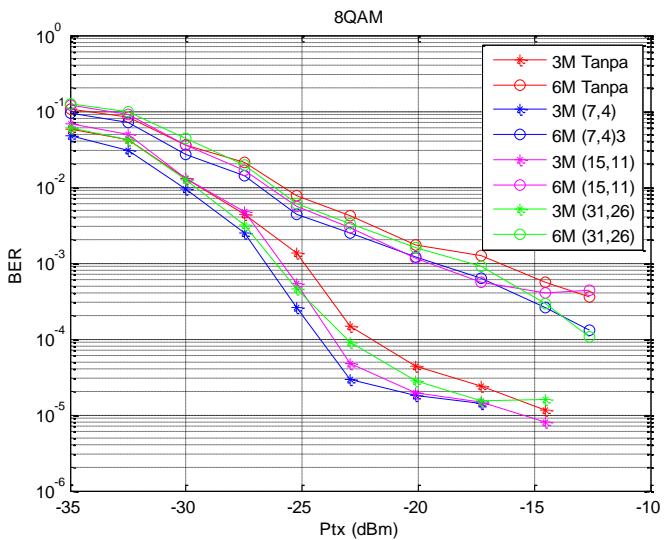
**Tabel 4. 9** Hasil Pengukuran dengan Kode Hamming(31, 26) Jarak 3 M

<b>PT<sub>x</sub> (dBm)</b>	<b>4 QAM</b>	<b>8 QAM</b>	<b>16 QAM</b>	<b>32 QAM</b>	<b>64 QAM</b>
-35	0,03670	0,06795	0,17510	0,33533	0,46570
-32.5	0,02013	0,04923	0,14235	0,31455	0,45303
-30	0,00294	0,01308	0,08206	0,24353	0,37092
-27.5	0,00021	0,00479	0,04736	0,20208	0,32114
-25	0,00001	0,00054	0,01315	0,14102	0,26908
-22.5	0,00000	0,00005	0,00660	0,08079	0,23482
-20	0,00000	0,00002	0,00189	0,04597	0,18274
-17.5	0,00000	0,00001	0,00028	0,02965	0,16113
-15	0,00000	0,00001	0,00001	0,01589	0,12186
-12.5	0,00000	0,00000	0,00000	0,01120	0,11500

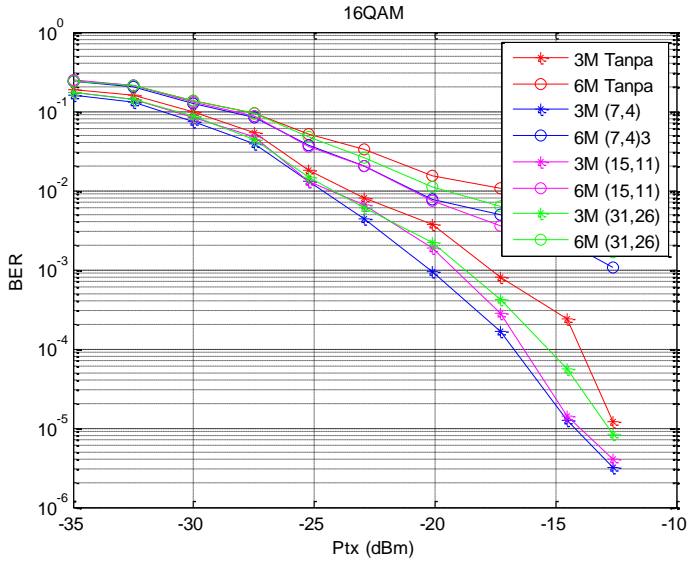
Menunjuk pada tabel hasil pengukuran pada jarak 6 meter dan 3 meter, dapat diperlihatkan nilai perbandingan BER dalam variasi jarak yang ditunjukkan pada Gambar 4.8.



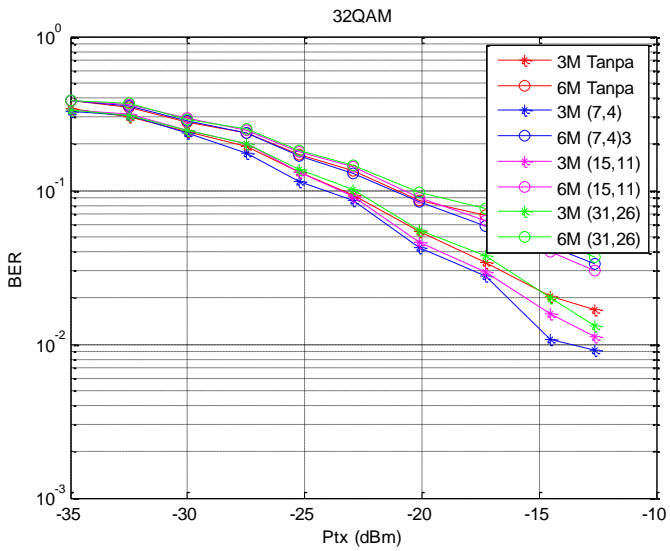
(a)



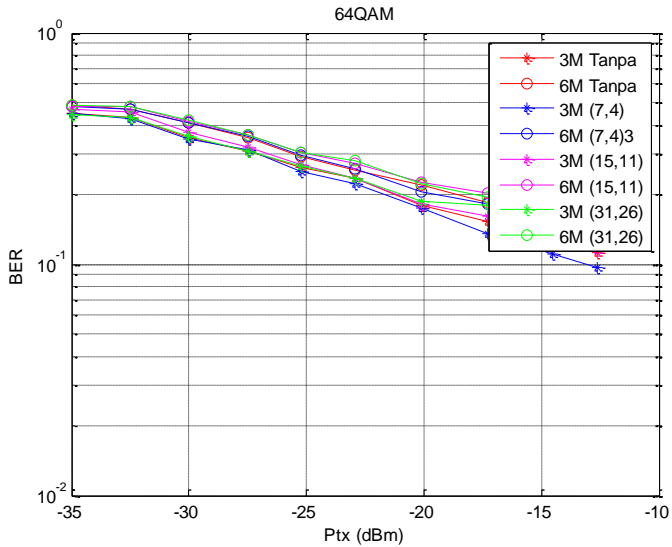
(b)



(c)



(d)



(e)

**Gambar 4. 8** BER Terhadap Daya Pancar pada Jarak 3 dan 6 Meter  
 (a)4-QAM (b)8-QAM (c)16-QAM (d)32-QAM (e)64-QAM

Pada grafik terlihat bahwa nilai BER mengalami kenaikan pada saat jarak 6 meter. Hal ini menunjukkan bahwa kualitas dari sistem komunikasi pada jarak 3 meter lebih baik. Selain itu daya yang diterima tidak sebesar pada saat transmisi pada jarak 3 meter. Karena sinyal yang ditransmisi sepanjang perjalanan mengalami redaman sehingga daya terima di penerima menjadi lemah.

*Halaman ini sengaja dikosongkan*



## **BAB V**

### **PENUTUP**

Pada bab 5 akan diuraikan mengenai kesimpulan dan saran berdasarkan hasil pengukuran maupun analisa data yang telah dilakukan sebelumnya.

#### **5.1 Kesimpulan**

Berdasarkan pembahasan dan analisa yang dilakukan pada Bab IV , maka dapat ditarik kesimpulan sebagai berikut:

- a. Kode Hamming dapat bekerja dengan baik pada level modulasi 4-QAM, 8-QAM dan 16-QAM. Tingginya level modulasi menyebabkan peluang *error* semakin besar, sehingga pada level modulasi 32-QAM dan 64-QAM, kinerja kode Hamming kurang maksimal karena kemampuan koreksinya terbatas hanya satu bit *error* pada setiap bloknya.
- b. Penggunaan Kode Hamming (7,4), (15,11) dan (31,26) pada modulasi 4-QAM jarak 6 meter mampu memberikan perbaikan dengan nilai secara berturut-turut kurang lebih 1.4 dB, 0.6 dB, dan 0.1 dB. Secara keseluruhan kode Hamming (7,4) menghasilkan nilai perbaikan yang lebih besar, namun efisiensi bandwidth yang digunakan lebih kecil seiring bertambahnya jumlah bit *parity* yang disisipkan.
- c. Dengan asumsi daya pancar dan kondisi yang sama, nilai BER pada jarak 3 meter lebih kecil dibandingkan pada jarak 6 meter.

#### **5.2 Saran**

Selama proses pengerjaan penelitian ini, saran yang bisa diajukan sebagai bahan untuk pengembangan lebih lanjut, yaitu:

- a. Penggunaan kode Hamming pada jenis modulasi yang lain
- b. Variasi informasi yang dikirimkan, seperti suara, *image* ataupun video.

*Halaman ini sengaja dikosongkan*

## LAMPIRAN A USULAN TUGAS AKHIR

Jurusan Teknik Elektro – FTI  
Institut Teknologi Sepuluh Nopember

### TE 141599 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Rina Asmoriwati  
Nomor Pokok : 2213105024  
Bidang Studi : Teknik Telekomunikasi Multimedia  
Tugas Diberikan : Semester Genap 2014/2015  
Dosen Pembimbing : 1. Dr. Ir. Suwadi, MT.  
2. Dr. Ir. Titiek Suryani, MT.

Judul Tugas Akhir : **Implementasi dan Evaluasi Kinerja Encoder-Decoder Hamming pada *M*-ary Quadrature Amplitude Modulation (M-QAM) Menggunakan *Wireless Open-Access Research Platform* (WARP)**  
*(Implementation and Performance Evaluation of Hamming Encoder-Decoder in *M*-ary Quadrature Amplitude Modulation (M-QAM) Using *Wireless Open-Access Research Platform* (WARP))*

12 FEB 2015

#### Uraian Tugas Akhir :

Salah satu kendala dalam sistem komunikasi digital adalah adanya *bit error* yang disebabkan oleh *noise* yang timbul selama proses transmisi berlangsung. Akibatnya data yang diterima pada penerima tidak sesuai dengan data yang dikirim. Apalagi jika modulasi yang digunakan adalah modulasi multilevel seperti *M*-ary Quadrature Amplitude Modulation (M-QAM). Semakin tinggi level modulasi maka tingkat kesalahannya pun semakin besar.

Kode Hamming merupakan salah satu *error control coding* yang berfungsi untuk mendeteksi kesalahan (*error detection*) sekaligus memperbaiki kesalahan (*error correction*) bit. Kode Hamming mampu untuk mengoreksi kesalahan tunggal dalam satu blok. Pada Tugas Akhir ini, kode Hamming (7,4), (15,11), dan (31,25) diimplementasikan pada M-QAM menggunakan modul *Wireless Open-Access Research Platform* (WARP) sebagai model pemancar dan penerima yang bersifat *real-time*. Evaluasi kinerja ditinjau dari nilai *Bit Error Rate* (BER) terhadap variasi ( $E_b/N_0$ ) untuk simulasi dan variasi daya pancar pengirim untuk pengukuran menggunakan WARP pada setiap variasi nilai *M*-ary QAM.

**Kata Kunci** : Hamming, M-QAM, WARP

Dosen Pembimbing I

**Dr. Ir. Suwadi, MT.**  
NIP. 19680818 1993031 002

Mengetahui,  
Jurusan Teknik Elektro FTI-JTS

Ketua,

**Dr. Tri Arief Sardjono, ST, MT.**  
NIP. 19700212 1995121 001

Dosen Pembimbing II

**Dr. Ir. Titiek Suryani, MT.**  
NIP. 19641130 1989032 001

Menyetujui,  
Bidang Studi Telekomunikasi  
Multimedia  
Koordinator

**Dr. Ir. Endroyo, DEA.**  
NIP. 19650404 1991021 001

*Halaman ini sengaja dikosongkan*

## LAMPIRAN B LISTING PROGRAM

### 1. Transmitter

```
%Random binary
n_bits = 100800;

%Proses Pembangkitan Data
rand('seed',1);
n_bin0 = randi([0 1],n_bits,1);

%Encoder
[h, g, n, kk] = hamngen(3);
n_bin = encode(n_bin0,n,kk,'linear/binary',g);
n_bits = length(n_bin);

%% Proses Konversi Simbol M-Ary
n_syms = n_bits/k;
n_sym = reshape(n_bin,n_syms,k);
n_dat = bi2de(n_sym,'left-msb');

%Baseband Modulation
n_sym_dat = modulate(h_bb_mod,n_dat);

%Hitung total data payload
Rsym = 3e6;
nsamp = floor(Fs/Rsym);
Rsym = Fs/nsamp;

%Pembentukan Preamble
pream = [0;0;0;0;-1+1i;1-1i;-1+1i;-1+1i;-1+1i;1-
1i;-1+1i;0;0;0;0;0;0;0;0];
pream_len = length(pream);

% Pembentukan matriks korelasi preamble
corr_win = 300;
pream_up = upsample(pream,nsamp);
ref_vect = zeros(corr_win,1);
ref_vect(1:length(pream_up)) = pream_up;
```

```

ref_matx =
toeplitz(ref_vect, circshift(ref_vect(corr_win:-
1:1),1));

% Transmission delay
txDelay = 400;

%Phase ambiguity detector
CSI_syms = 150
CSI_dat = ones(CSI_syms,1);

%Desain Square Root Raised Cosine
shape = 'Square Root Raised Cosine';
N_Ord = 6; %Genap
R_SRRC = 0.9; %[0-1]
SRRC_spec =
fdesign.pulseshaping(nsamp, shape, 'Nsym, Beta', N_Ord, R_SRRC);
h_SRRC = design(SRRC_spec);
flt_delay = h_SRRC.order/2; %or nsamp*N_Ord
norm_factor = max(h_SRRC.numerator);
h_SRRC.numerator = h_SRRC.numerator/norm_factor;
%Normalisasi filter

%Jumlah simbol maksimum
data_sym = floor((L-(pream_len*nsamp)-txDelay-flt_delay)/nsamp);

%Bit stuffing and frame construction
%Tentukan jumlah simbol data diantara simbol CSI
sym_size = floor(data_sym/(CSI_syms-1))-1;
%Tentukan total sub-frame data
sub_frame = CSI_syms-1;
%Total panjang frame aktual dengan stuffed bits
fr_len = sub_frame*sym_size + CSI_syms;
%Update nilai data_sym
data_sym = fr_len - CSI_syms;
%Hitung total frame data, hitung padding,
tambahkan bit stuffing
n_frames = ceil(n_syms/data_sym);
stuff_sym = [upsample(CSI_dat(1:CSI_syms-1), sym_size+1); CSI_dat(end)];

```

```

n_dat_fr = repmat(stuff_sym,1,n_frames);
pad = n_frames*data_sym - n_syms;
%Tambahkan zero padding pada symbol
n_sym_dat = [n_sym_dat;zeros(pad,1)];

%Reshape n_sym_dat
n_sym_dat =
reshape(n_sym_dat,data_sym,n_frames);

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

%Tambahkan preamble
temp_mat1 = repmat(pream,1,n_frames);
xt_dat = [temp_mat1;n_dat_fr];

%Interpolasi, matched filtering, tambahkan
filter delay
xt_mf_bb =
filter(h_SRRRC,[upsample(xt_dat,nsamp);zeros(flt_
delay,n_frames)]);
xt_mf = zeros(size(xt_mf_bb));

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

```

## 2. Listing Program untuk Receiver

```
%Sinkronisasi dan Downconvert semua frame
time = ((0:length(rx_IQ)-1)/Fs).';
rx_IQ_disp = rx_IQ;
for ii=1:n_frames
%vco_sig = syn_frame(rx_IQ(:,ii),f,Fs);
    vco_sig = exp(1i*2*pi*f*time);
    rx_IQ(:,ii) = rx_IQ(:,ii).*conj(vco_sig);
end

%Sinkronisasi fasa
%Matched filtering
hrx_SRRC = design(SRRC_spec);
hrx_SRRC.numerator = hrx_SRRC.numerator *
(norm_factor*nsamp);
yt_mf =
filter(hrx_SRRC,[rx_IQ;zeros(flt_delay,n_frames)
]);

yt_rx_data = zeros(size(n_dat_fr));
z_sym = zeros(size(n_sym_dat));
ch_est = zeros(size(n_dat_fr));

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

%Cari korelasi tertinggi
    correlation = abs( (temp_vct1(1:corr_win)')
* ref_matx );
    pream_start = find(correlation ==
max(correlation));
    first_sample = pream_start+length(pream_up);

%Downsampling
    temp_vct2 = temp_vct1(first_sample:end);
%Start sampling at initial point
    temp_vct3 =
downsample(temp_vct2(1:end),nsamp);
```



```

%Trim
    temp_vct3 = temp_vct3(1:fr_len);
%Save trimmed vector
    yt_rx_data(:,ii) = temp_vct3;

%Estimasi kanal
    ch_est(:,ii) =
estimate_chan(yt_rx_data(:,ii),sym_size);
%Received symbols:
    temp_vct3 =
yt_rx_data(:,ii).*conj(ch_est(:,ii));
    multip = [abs(ch_est(:,ii)).^2];
%    multip = sum(multip,2);
    yt_rx_data(:,ii) = (temp_vct3)./multip;

%Symbol de-stuffing
    temp_vct3 =
sym_destuffing(yt_rx_data(:,ii),sym_size);
    z_sym(:,ii) = temp_vct3;
end

%Ubah yt_rx_data menjadi vektor, potong yt_syms
yt_syms = reshape(z_sym,data_sym*n_frames,1);
yt_syms = yt_syms(1:length(n_dat));

%Demodulasi
z_sym = demodulate(h_bb_demod,yt_syms);

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

%% Decoder
yy= de2bi(rx_asiso,'left-msb');
rx_bit =reshape(yy,1,[]);
rx_bit0 = rx_bit .';
rx_bit = decode(rx_bit,n,kk,'hamming/binary')
.';

%% BIT ERROR RATE
[num_err2 BER_DENGAN] = biterr(n_bin0,rx_bit)

```

## LAMPIRAN A USULAN TUGAS AKHIR

Jurusan Teknik Elektro – FTI  
Institut Teknologi Sepuluh Nopember

### TE 141599 TUGAS AKHIR – 4 SKS

Nama Mahasiswa : Rina Asmoriwati  
Nomor Pokok : 2213105024  
Bidang Studi : Teknik Telekomunikasi Multimedia  
Tugas Diberikan : Semester Genap 2014/2015  
Dosen Pembimbing : 1. Dr. Ir. Suwadi, MT.  
2. Dr. Ir. Titiek Suryani, MT.

Judul Tugas Akhir : **Implementasi dan Evaluasi Kinerja Encoder-Decoder Hamming pada *M*-ary Quadrature Amplitude Modulation (M-QAM) Menggunakan *Wireless Open-Access Research Platform* (WARP)**  
*(Implementation and Performance Evaluation of Hamming Encoder-Decoder in *M*-ary Quadrature Amplitude Modulation (M-QAM) Using *Wireless Open-Access Research Platform* (WARP))*

12 FEB 2015

#### Uraian Tugas Akhir :

Salah satu kendala dalam sistem komunikasi digital adalah adanya *bit error* yang disebabkan oleh *noise* yang timbul selama proses transmisi berlangsung. Akibatnya data yang diterima pada penerima tidak sesuai dengan data yang dikirim. Apalagi jika modulasi yang digunakan adalah modulasi multilevel seperti *M*-ary Quadrature Amplitude Modulation (M-QAM). Semakin tinggi level modulasi maka tingkat kesalahannya pun semakin besar.

Kode Hamming merupakan salah satu *error control coding* yang berfungsi untuk mendeteksi kesalahan (*error detection*) sekaligus memperbaiki kesalahan (*error correction*) bit. Kode Hamming mampu untuk mengoreksi kesalahan tunggal dalam satu blok. Pada Tugas Akhir ini, kode Hamming (7,4), (15,11), dan (31,25) diimplementasikan pada M-QAM menggunakan modul *Wireless Open-Access Research Platform* (WARP) sebagai model pemancar dan penerima yang bersifat *real-time*. Evaluasi kinerja ditinjau dari nilai *Bit Error Rate* (BER) terhadap variasi ( $E_b/N_0$ ) untuk simulasi dan variasi daya pancar pengirim untuk pengukuran menggunakan WARP pada setiap variasi nilai *M*-ary QAM.

**Kata Kunci** : Hamming, M-QAM, WARP

Dosen Pembimbing I

**Dr. Ir. Suwadi, MT.**  
NIP. 19680818 1993031 002

Mengetahui,  
Jurusan Teknik Elektro FTI-JTS

Ketua,

**Dr. Tri Arief Sardjono, ST, MT.**  
NIP. 19700212 1995121 001

Dosen Pembimbing II

**Dr. Ir. Titiek Suryani, MT.**  
NIP. 19641130 1989032 001

Menyetujui,  
Bidang Studi Telekomunikasi  
Multimedia  
Koordinator

**Dr. Ir. Endroyo, DEA.**  
NIP. 19650404 1991021 001

*Halaman ini sengaja dikosongkan*

## LAMPIRAN B LISTING PROGRAM

### 1. Transmitter

```
%Random binary
n_bits = 100800;

%Proses Pembangkitan Data
rand('seed',1);
n_bin0 = randi([0 1],n_bits,1);

%Encoder
[h, g, n, kk] = hammggen(3);
n_bin = encode(n_bin0,n,kk,'linear/binary',g);
n_bits = length(n_bin);

%% Proses Konversi Simbol M-Ary
n_syms = n_bits/k;
n_sym = reshape(n_bin,n_syms,k);
n_dat = bi2de(n_sym,'left-msb');

%Baseband Modulation
n_sym_dat = modulate(h_bb_mod,n_dat);

%Hitung total data payload
Rsym = 3e6;
nsamp = floor(Fs/Rsym);
Rsym = Fs/nsamp;

%Pembentukan Preamble
pream = [0;0;0;0;-1+1i;1-1i;-1+1i;-1+1i;-1+1i;1-
1i;-1+1i;0;0;0;0;0;0;0;0];
pream_len = length(pream);

% Pembentukan matriks korelasi preamble
corr_win = 300;
pream_up = upsample(pream,nsamp);
ref_vect = zeros(corr_win,1);
ref_vect(1:length(pream_up)) = pream_up;
```

```

ref_matx =
toeplitz(ref_vect, circshift(ref_vect(corr_win:-
1:1),1));

% Transmission delay
txDelay = 400;

%Phase ambiguity detector
CSI_syms = 150
CSI_dat = ones(CSI_syms,1);

%Desain Square Root Raised Cosine
shape = 'Square Root Raised Cosine';
N_Ord = 6; %Genap
R_SRRC = 0.9; %[0-1]
SRRC_spec =
fdesign.pulseshaping(nsamp, shape, 'Nsym, Beta', N_Ord, R_SRRC);
h_SRRC = design(SRRC_spec);
flt_delay = h_SRRC.order/2; %or nsamp*N_Ord
norm_factor = max(h_SRRC.numerator);
h_SRRC.numerator = h_SRRC.numerator/norm_factor;
%Normalisasi filter

%Jumlah simbol maksimum
data_sym = floor((L-(pream_len*nsamp)-txDelay-flt_delay)/nsamp);

%Bit stuffing and frame construction
%Tentukan jumlah simbol data diantara simbol CSI
sym_size = floor(data_sym/(CSI_syms-1))-1;
%Tentukan total sub-frame data
sub_frame = CSI_syms-1;
%Total panjang frame aktual dengan stuffed bits
fr_len = sub_frame*sym_size + CSI_syms;
%Update nilai data_sym
data_sym = fr_len - CSI_syms;
%Hitung total frame data, hitung padding,
tambahkan bit stuffing
n_frames = ceil(n_syms/data_sym);
stuff_sym = [upsample(CSI_dat(1:CSI_syms-1), sym_size+1); CSI_dat(end)];

```

```

n_dat_fr = repmat(stuff_sym,1,n_frames);
pad = n_frames*data_sym - n_syms;
%Tambahkan zero padding pada symbol
n_sym_dat = [n_sym_dat;zeros(pad,1)];

%Reshape n_sym_dat
n_sym_dat =
reshape(n_sym_dat,data_sym,n_frames);

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

%Tambahkan preamble
temp_mat1 = repmat(pream,1,n_frames);
xt_dat = [temp_mat1;n_dat_fr];

%Interpolasi, matched filtering, tambahkan
filter delay
xt_mf_bb =
filter(h_SRRRC,[upsample(xt_dat,nsamp);zeros(flt_
delay,n_frames)]);
xt_mf = zeros(size(xt_mf_bb));

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

```

## 2. Listing Program untuk Receiver

```
%Sinkronisasi dan Downconvert semua frame
time = ((0:length(rx_IQ)-1)/Fs).';
rx_IQ_disp = rx_IQ;
for ii=1:n_frames
%vco_sig = syn_frame(rx_IQ(:,ii),f,Fs);
    vco_sig = exp(1i*2*pi*f*time);
    rx_IQ(:,ii) = rx_IQ(:,ii).*conj(vco_sig);
end

%Sinkronisasi fasa
%Matched filtering
hrx_SRRC = design(SRRC_spec);
hrx_SRRC.numerator = hrx_SRRC.numerator *
(norm_factor*nsamp);
yt_mf =
filter(hrx_SRRC,[rx_IQ;zeros(flt_delay,n_frames)
]);

yt_rx_data = zeros(size(n_dat_fr));
z_sym = zeros(size(n_sym_dat));
ch_est = zeros(size(n_dat_fr));

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

%Cari korelasi tertinggi
    correlation = abs( (temp_vct1(1:corr_win)')
* ref_matx );
    pream_start = find(correlation ==
max(correlation));
    first_sample = pream_start+length(pream_up);

%Downsampling
    temp_vct2 = temp_vct1(first_sample:end);
%Start sampling at initial point
    temp_vct3 =
downsample(temp_vct2(1:end),nsamp);
```

```

%Trim
    temp_vct3 = temp_vct3(1:fr_len);
%Save trimmed vector
    yt_rx_data(:,ii) = temp_vct3;

%Estimasi kanal
    ch_est(:,ii) =
estimate_chan(yt_rx_data(:,ii),sym_size);
%Received symbols:
    temp_vct3 =
yt_rx_data(:,ii).*conj(ch_est(:,ii));
    multip = [abs(ch_est(:,ii)).^2];
%    multip = sum(multip,2);
    yt_rx_data(:,ii) = (temp_vct3)./multip;

%Symbol de-stuffing
    temp_vct3 =
sym_destuffing(yt_rx_data(:,ii),sym_size);
    z_sym(:,ii) = temp_vct3;
end

%Ubah yt_rx_data menjadi vektor, potong yt_syms
yt_syms = reshape(z_sym,data_sym*n_frames,1);
yt_syms = yt_syms(1:length(n_dat));

%Demodulasi
z_sym = demodulate(h_bb_demod,yt_syms);

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

%% Decoder
yy= de2bi(rx_siso,'left-msb');
rx_bit =reshape(yy,1,[]);
rx_bit0 = rx_bit .';
rx_bit = decode(rx_bit,n,kk,'hamming/binary')
.';

%% BIT ERROR RATE
[num_err2 BER_DENGAN] = biterr(n_bin0,rx_bit)

```



## DAFTAR PUSTAKA

- [1] WARP Project-Wireless Open Access Research Platform, “*About the WARP Project*” <URL: <http://warpproject.org/trac/wiki/about>>, [Disunting: Februari 2, 2015].
- [2] Darwis, Rizadi Sasmita. “*Manual Penggunaan Modul Wireless Open-Access Research Platform WARP*”. Workshop Software Defined Radio dan WARP. Institut Teknologi Sepuluh Nopember (ITS). Surabaya. 2014.
- [3] Darwis, Rizadi Sasmita. “*Implementasi Sistem Komunikasi SISO dan MIMO 2x2 pada Platform WARP*”. Tesis Magister Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS). Surabaya. 2014.
- [4] Hartanto, Sri. “*Pengurangan Bit Error pada Modulasi M-QAM dalam Kanal Rayleigh Fading dengan Teknik Spatial Multiplexing dan Metode ZF-SIC, MMSE-SIC, Maximum Likelihood*”. Tesis Magister Teknik Elektro, Universitas Indonesia. Depok. 2010.
- [5] Kisiel, K. Sahota, D. Swaminathan, G. “*Quadrature Amplitude Modulation : A Simulation Study*”. <URL: <http://citeseerx.psu.edu>>, [Disunting: Januari 17, 2015].
- [6] Lin, Shu. Costello, Daniel J. “*Error Control Coding: Fundamentals and Applications*”. Prentice Hall. United State of America. 1983.
- [7] Malek, Massoud. “*Coding Theory*”. Lecture Note. California State University. East Bay.
- [8] Sklar, Bernard. “*Digital Communications Fundamentals and Application*”. Prentice Hall. California. 2001.

## BIODATA PENULIS



Rina Asmorowati dilahirkan di Ponorogo pada tanggal 20 April 1991. Penulis merupakan anak pertama dari pasangan Sarno dan Umiyatun. Penulis memulai pendidikan formal di SDN Manuk Siman Ponorogo dari tahun 1997-2003, SMPN 2 Ponorogo pada tahun 2003-2006, SMAN 1 Ponorogo pada tahun 2006-2009. Setelah lulus SMA penulis melanjutkan studi Diploma III (DIII) di Politeknik Negeri Malang, Jurusan Teknik Elektro, Program Studi Teknik Telekomunikasi pada tahun 2009-2012 dan pada tahun 2013 penulis melanjutkan studi strata 1 (S1) di Institut Teknologi Sepuluh Nopember, Fakultas Teknologi Industri, Jurusan Teknik Elektro Bidang Studi Telekomunikasi Multimedia. Penulis dapat dihubungi melalui alamat email [rinaasmorowati@gmail.com](mailto:rinaasmorowati@gmail.com).