

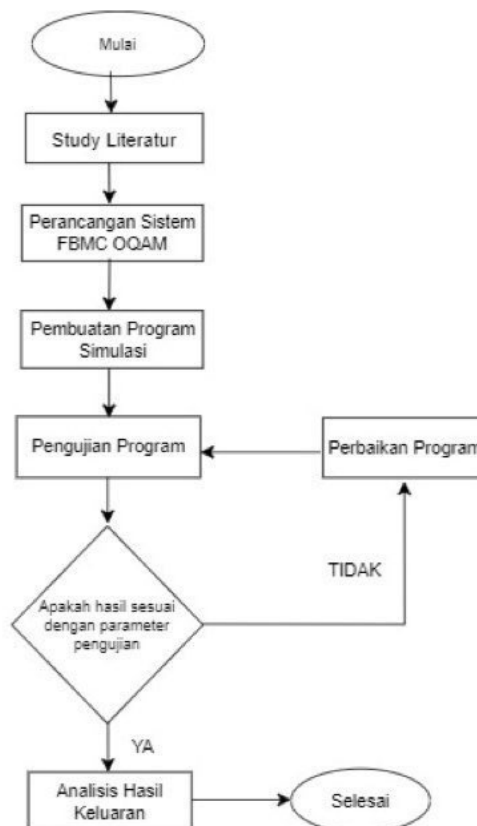
## BAB 3 METODE PENELITIAN

### 3.1 ALAT YANG DIGUNAKAN

Penelitian ini menggunakan suatu pemodelan dalam pengelompokan data *K-Means* sesuai dengan karakteristiknya untuk menganalisis perbaikan BER. Model simulasi yang diimplementasikan dalam penelitian ini, menggunakan sebuah program MATLAB R2016b. Untuk spesifikasi *device* laptop yang digunakan yaitu AMD *Radeon* dengan kecepatan CPU 2 GHz dan RAM 8 GB.

### 3.2 ALUR PENELITIAN

Penelitian dilakukan dalam beberapa tahapan yaitu tahap perancangan program, tahap pembuatan simulasi, tahap pengujian simulasi, dan tahap terakhir adalah analisis hasil pengujian simulasi menggunakan *Matlab*, seperti terlihat pada Gambar 3.1 berikut ini:



**Gambar 3.1** *Flowchart* Tahapan Penelitian

Berdasarkan Gambar 3.1 menjelaskan rancangan simulasi program yang dilakukan. Pada bagian pertama yaitu *study literatur yang* bertujuan untuk mengumpulkan informasi untuk penelitian ini. Selanjutnya perancangan sistem FBMC OQAM yaitu membuat konsep yang akan dilakukan. Pada penelitian ini pembuatan program simulasi menggunakan program MATLAB. Kemudian setelah pembuatan program dilakukan pengujian program dengan variasi level modulasi berdasarkan parameter *Bit Error Rate (BER)*, *Signal to Noise Ratio (SNR)* dan kapasitas kanal. Jika hasil keluran tidak sesuai dengan parameter maka dilakukan perbaikan program dan pengujian program. Sebaliknya, jika hasil keluran sesuai dengan parameter maka hasil akan dilakukan analisis hasil keluran.

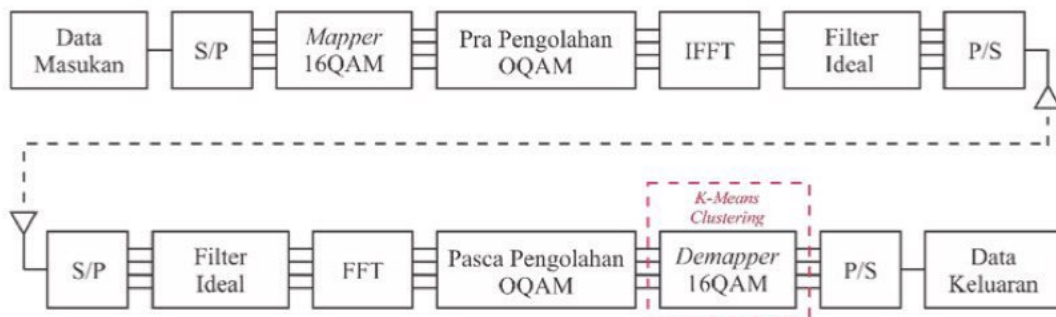
### 3.3 PARAMETER SIMULASI

Parameter yang digunakan dalam simulasi ini, sebagaimana tertera pada Tabel 3.1.

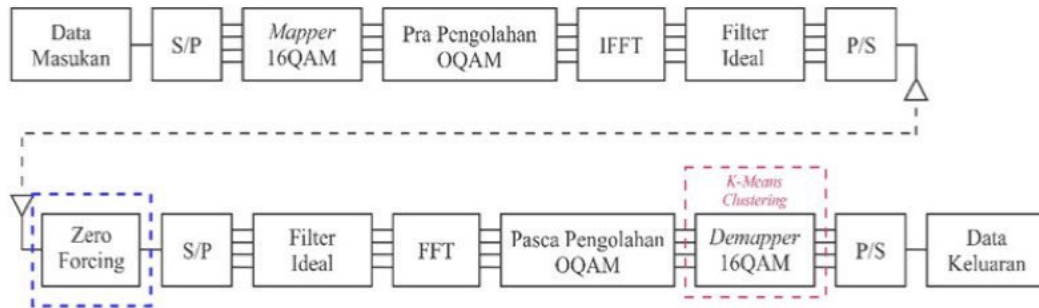
Tabel 3. 1 Parameter Simulasi

Simbol	Parameter	Nilai
$n_{tx}$	jumlah antenna pengirim	1
$n_{rx}$	Jumlah antenna penerima	1
m	Modulasi	16 QAM
m1	Jumlah level modulasi	4
$f_s$	Frekuensi cuplik	22050 Hz

### 3.4 PERMODELAN SISTEM FBMC OQAM 16 QAM



**Gambar 3. 2 Bagan FBMC OQAM menggunakan *K-Means Clustering* sebagai *Demapper***



**Gambar 3. 3 Bagan FBMC OQAM Zero Forcing menggunakan K-Means Clustering sebagai Demapper**

Berikut penjelasan dari bagan FBMC diatas, yaitu :

### Sisi Pengirim :

#### 1. Data Masukan

Data masukan dalam simulasi ini berupa sinyal audio dalam bentuk *file \*.wav*. Sinyal *audio* memiliki amplitudo yang ternormalisasi dari -1Volt hingga 1Volt. Pengubahan sinyal ke dalam bilangan biner dengan tujuan agar sinyal bisa ditransmisikan. Persamaan untuk pengubahan sinyal ke dalam biner adalah :

$$V_1 = V_0 + 1$$

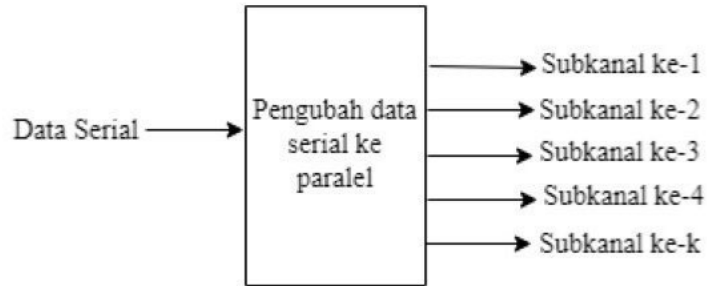
$$V_t = V_1 * 2^{nbit-1}$$

$$B = \text{binary} (Vt)$$

Sehingga amplitudo tersebut perlu ditambah 1 agar menjadi positif. Setelah itu dikalikan dengan  $2^{n-1}$  untuk dapat mengubahnya menjadi bilangan desimal, dimana  $n$  merupakan jumlah *bit* biner pada setiap titik sinyal *audio* yakni berjumlah 8 *bit* biner. Nilai desimal itulah yang akan diubah menjadi bentuk biner.

#### 2. Pengubahan Data Serial ke Pararel (S/P)

Pada blok diagram pengubah seri ke pararel ini berfungsi untuk mengubah bentuk dari data bit seri kedalam bentuk paralel dengan mengelompokkan bit tersebut sesuai dengan tingkatan modulasi yang digunakan.



**Gambar 3. 4** Bagan pengubah data serial ke paralel

### 3. Mapper 16-QAM

Hasil keluaran dari pengubah data serial ke paralel. Selanjutnya dipetakan dengan modulasi dasar yaitu 16 QAM. Pemetaan 16 QAM dilakukan untuk mengubah data masukan binary menjadi bentuk bilangan kompleks  $S_k = I_k + JQ_k$ , I adalah *inphase* (bilangan real), k adalah variabel simbol, dan Q adalah bilangan imajiner (*quadrature*). Setiap simbol yang diterima pada pemetaan ini terdiri dari empat bit. Agar dapat diterima pada kanal *wireless* proses dimodulasikan menggunakan modulasi QAM.

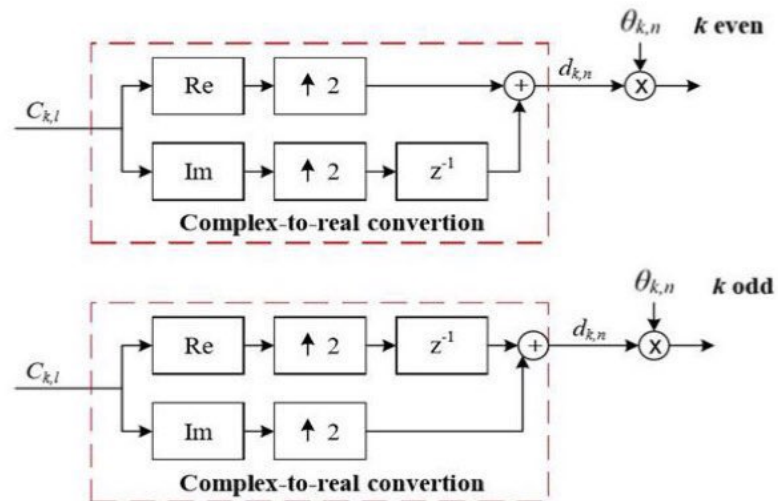
Tabel 3. 2 Pemetaan 16 QAM

No.	Biner 4 bit				Bilangan Kompleks (x)	
	Bit-4	Bit-3	Bit-2	Bit-1	Real	Imajiner
1	0	0	0	0	0,3162	0,3162 i
2	0	0	0	1	-0,3162	0,3162 i
3	0	0	1	0	0,3162	-0,3162 i
4	0	0	1	1	-0,3162	-0,3162 i
5	0	1	0	0	0,9486	0,3162 i
6	0	1	0	1	-0,9486	0,3162 i
7	0	1	1	0	0,9486	-0,3162 i
8	0	1	1	1	-0,9486	-0,3162 i
9	1	0	0	0	0,3162	0,9486 i
10	1	0	0	1	-0,3162	0,9486 i
11	1	0	1	0	0,3162	-0,9486 i
12	1	0	1	1	-0,3162	-0,9486 i
13	1	1	0	0	0,9486	0,9486 i

14	1	1	0	1	-0,9486	0,9486 i
15	1	1	1	0	0,9486	-0,9486 i
16	1	1	1	1	-0,9486	-0,9486 i

#### 4. Pra Pengolahan OQAM

Pada proses pra pengolahan OQAM ini menerima *input*-an dari pemeta 16-QAM yang kemudian akan dilakukan operasi *staggering*. Operasi ini merubah bilangan kompleks menjadi bilangan riil, dimana bagian riil dan imajiner dari data simbol bernilai kompleks dipisahkan kembali untuk membentuk dua simbol baru. Urutan simbol-simbol baru ini tergantung dari nomor *sub* saluran yang dibedakan menjadi saluran bernomor genap dan ganjil. Pra pengolahan OQAM ditunjukkan pada Gambar 3.5.



**Gambar 3. 5** Pra Pengolahan OQAM untuk saluran ganjil dan genap

#### 5. Inverse Fast Fourier Transform (IFFT)

Metode IFFT adalah *inverse* atau kebalikan dari *Fast Fourier Transform* (FFT), yang mana FFT merupakan metode untuk pemecahan sinyal diskret. Sedangkan untuk *Inverse Fast Fourier Transform* (IFFT) adalah proses untuk mengubah sinyal dari domain frekuensi ke domain waktu. *Subcarrier* pada IFFT memiliki frekuensi harmonisasi kelipatan bulat dari frekuensi dasarnya seperti halnya komponen deret *fourier* pada sinyal komposit.

## 6. Filter Ideal

Simbol keluaran dari IFFT dilewatkan melalui filter ideal. Sehingga simbol yang ada langsung diteruskan tanpa adanya data *noise* yang dihilangkan.

## 7. Pengubah Pararel ke Serial (P/S)

Pengubah pararel ke serial di sisi pengirim, berfungsi sebagai mengubah keluaran dari filter ideal yang semula bentuknya paralel menjadi serial, hal itu bertujuan untuk dapat dikirimkan menjadi 1 baris.

## 8. Pemodelan Kanal Transmisi

Kanal transmisi yang digunakan pada simulasi ini adalah kanal *Additive White Gaussian Noise* (AWGN). Pada kanal AWGN diasumsikan memiliki derau yang terdistribusi normal (*Gaussian*). Derau AWGN terdistribusi normal dengan nilai rata-rata adalah nol. Sesuai dengan namanya, derau ini memiliki sifat *additive* yang berarti derau ini akan dijumlahkan dengan sinyal informasi, sifat *white* dimana derau ini memiliki rapat daya yang konstan, tidak bergantung pada frekuensi operasi sistem, dan mengandung komponen frekuensi. Sifat *gaussian* berarti nilai tegangan derau terdistribusi normal.

### **Sisi Penerima :**

#### 1. Deteksi Simbol Pemaksaan Nol (*Zero Forcing*)

Proses deteksi simbol dilakukan untuk memperoleh sinyal secara digital untuk mendapatkan data nilai sinyal aslinya. Pada simulasi ini difokuskan pada algoritma yang paling sederhana yaitu *Zero Forcing* (ZF). Algoritma ini dilakukan karena sinyal-sinyal yang diterima oleh antenna penerima merupakan hasil dari perkalian sinyal-sinyal dari antenna pengirim dengan jenis kanal yang digunakan dan kemudian ditambahkan dengan derau. Algoritma ini cukup mudah digunakan dimana sinyal yang diterima dikalikan dengan dikembalikan dari kanal H, kemudian ditambahkan dengan derau AWGN.

2. Pengubah Data Serial menjadi Paralel (S/P)

Pengubah data serial menjadi paralel di sisi penerima berfungsi untuk mengubah masukan dari antenna penerima berbentuk serial menjadi bentuk paralel.

3. Filter Ideal

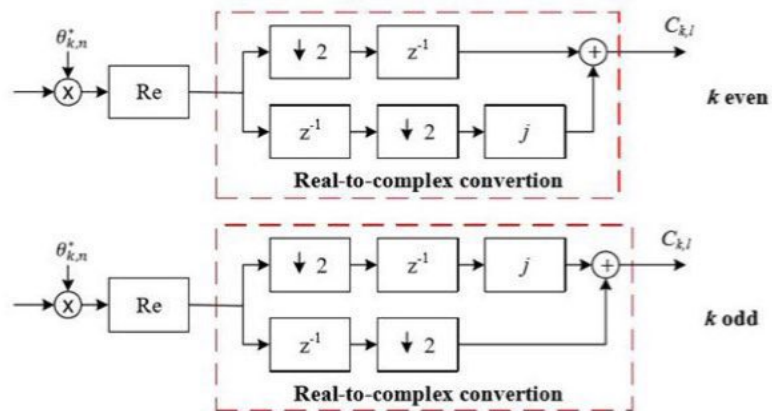
Simbol hasil keluaran S/P akan dikenakan filter ideal, simbol yang ada diteruskan secara langsung tanpa data yang dihilangkan.

4. *Fast Fourier Transform* (FFT)

Keluaran dari filter ideal dikalikan dengan proses transformasi *fourier* (FFT) yang berfungsi untuk memisahkan kembali data informasi dengan sinyal *carrier*. Proses ini dapat juga disebut dengan proses demodulasi *multicarrier*. Dalam proses ini sinyal dalam domain waktu diubah kembali ke domain frekuensi.

5. Pasca Pengolahan OQAM

Blok pasca pengolahan OQAM merupakan kebalikan dari proses pra pengolahan OQAM. Seperti yang ditunjukkan pada Gambar 3.6, operasi pertama penggandaan demi urutan, mengalikan dengan *conjugate* atau kebalikan dari  $\theta_{k,n} = j^{(k+n)}$  untuk menghasilkan nilai riil. Kemudian dilanjutkan dengan operasi *dis-staggering* yang merubah bilangan riil menjadi bilangan kompleks, dimana dua simbol bernilai riil akan mengalami proses penurunan laju sampel sebesar 2 kali (*downsampling* atau  $\downarrow 2$ ). Kemudian mengalami penundaan 1 simbol ( $z^{-1}$ ) dan salah satu dari bilangan riil dan imajiner harus ada yang dikalikan dengan simbol  $j$  untuk menghasilkan bilangan kompleks.



Gambar 3. 6 Pasca Pengolahan OQAM [16].

#### 6. *K-Means Clustering* sebagai *Demapper* 16 QAM

Pada bagian penerima, terjadi proses kebalikan yaitu dari pemeta 16 QAM yang merupakan proses pengawa peta atau *demapper* 16 QAM. Pada proses ini setiap simbol akan dipetakan kembali untuk membentuk empat data bit informasi kembali. Proses pengambilan keputusan pada pengawa petaan 16 QAM, digunakan untuk menentukan simbol mana yang sebenarnya dikirimkan oleh antenna pengirim. Pengambilan keputusan yang dilakukan adalah menggunakan metode *K-Means Clustering*, hal ini diperlukan karena adanya pengaruh kanal dan derau yang mengakibatkan simbol QAM di bagian penerima menjadi tidak seperti data yang dikirimkan dari sisi pengirim yang didapat pada keluaran pemeta 16 QAM pada sisi pengirim. Kemudian hasil pengelompokkan *K-Means Clustering* ini akan direpresentasikan menjadi bit-bit informasi berbentuk paralel. Adapun aturan pengambilan keputusan seperti pada Tabel 3.3 di bawah ini:

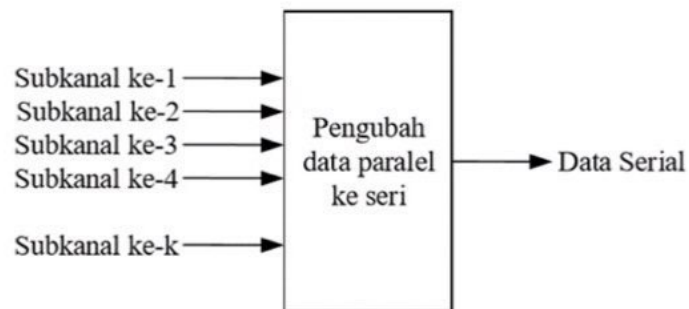
Tabel 3.3 *Demapper* 16 QAM

No.	Riil(x)	Imajiner(y)	Keluaran
1	$0,6324 > x > 0$	$0 < y < 0,6324$	0 0 0 0
2	$-0,6324 > x > 0$	$0 < y < 0,6324$	0 0 0 1
3	$0,6324 > x > 0$	$0 < y < -0,6324$	0 0 1 0
4	$-0,6324 > x > 0$	$0 < y < -0,6324$	0 0 1 1
5	$x < 0,6324$	$0 < y < 0,6324$	0 1 0 0
6	$x < -0,6324$	$0 < y < 0,6324$	0 1 0 1
7	$x > 0,6324$	$0 < y < -0,6324$	0 1 1 0
8	$x < -0,6324$	$-0,6324 < y < 0$	0 1 1 1
9	$0,6324 > x > 0$	$y > 0,6324$	1 0 0 0
10	$-0,6324 < x < 0$	$y > 0,6324$	1 0 0 1
11	$0,6324 > x > 0$	$y < -0,6324$	1 0 1 0
12	$-0,6324 < x < 0$	$y < -0,6324$	1 0 1 1
13	$x < 0,6324$	$y > 0,6324$	1 1 0 0
14	$x < -0,6324$	$y > 0,6324$	1 1 0 1
15	$x > 0,6324$	$y < -0,6324$	1 1 1 0
16	$x < -0,6324$	$y < -0,6324$	1 1 1 1



### 7. Pengubah Paralel menjadi Serial (P/S)

Blok pengubah paralel menjadi serial berfungsi untuk mengubah bentuk bit yang semula berbentuk bit paralel menjadi bit serial. Dengan mengelompokkan empat *bit binary* keluaran dari *demapper* 16 QAM, menjadi satu baris sebagai keluaran dari proses pengawapetaan 16-QAM ke data *serial bit*. Pengubah P/S ditunjukkan pada Gambar 3.7.



**Gambar 3. 7 Pengubah Data Paralel menjadi Serial**

### 8. Data Keluaran

Di sisi penerima, bit-bit biner tersebut akan dikembalikan ke bentuk semula yaitu sinyal informasi berupa audio, yang mana hasilnya dapat digunakan untuk membandingkan bit yang diterima oleh *receiver* dengan bit yang dikirimkan pada sisi *transmitter* untuk mendapatkan nilai *Bit Error Ratio* (BER). perhitungan BER dilakukan dengan membandingkan total bit yang *error* dengan total bit yang dikirim sesuai dengan persamaan (2.5).