

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [6] membahas mengenai perbandingan teknik *multicarrier* FBMC-OQAM dan sistem komunikasi OFDM. Sistem *Frequency Division Orthogonal Multiplexing* (OFDM) telah memantapkan diri sebagai salah satu teknologi *multicarrier* terkemuka dalam komunikasi nirkabel 4G. Tujuan penelitian ini adalah untuk mengatasi kelemahan OFDM dengan menggunakan teknik *multicarrier* FBMC (*Filter Bank Multicarrier*). Hasil kinerjanya membuatnya lebih kompatibel untuk komunikasi nirkabel 5G. Hal tersebut dikarenakan *Power Spectral Density* (PSD) yang dihasilkan lebih tajam, kurva kinerja *Bit Error Rate* (BER) lebih baik, dan *Peak to Average Power Ratio* (PAPR) yang dihasilkan lebih rendah untuk membuang *Cyclic Prefix* (CP) dengan bantuan *Filter Bank*. Untuk meminimalisir *Inter-Symbol Interference* (ISI) dan *Inter-Carrier Interference* (ICI), penelitian ini memfokuskan kinerja pada OQAM, frekuensi penyebaran dan skalabilitas IFFT atau FFT dan menunjukkan cara *bank filter* meningkatkan kinerja sistem FBMC. Hasilnya menunjukkan simulasi menggunakan FBMC lebih efisien daripada metode OFDM. Hal tersebut dibuktikan dari nilai *Out Of Band* (OOB) FBMC menurun dari -31 dB menjadi -58 dB. Selain itu, kinerja FBMC 16-QAM dalam hal nilai BER (*Bit Error Rate*) menurun dari 27 dB menjadi 20 dB, dan nilai 4-QAM FBMC hampir lebih rendah sebesar 5 dB dari 4-QAM OFDM. Sedangkan untuk kinerja *Peak to Average Power Ratio* (PAPR) pada OFDM memiliki kurva yang menurun daripada FBMC (16 dB ke 12 dB).

Penelitian [7] membahas tentang adanya kekurangan dari metode *K-Means Clustering* yaitu, bergantung pada posisi pusat awal yang mana titik tersebut dipilih secara acak sehingga dapat menghasilkan kualitas pengelompokan yang buruk. Penelitian ini mengoptimalkan metode *K-Means* dengan 3 prinsip, pertama optimalisasi pada hirarkis, K^* sebagai pusat awal untuk kluster dengan $K^* > K$. Artinya kluster kecil mendapatkan pusat awal sebanyak mungkin. Hal ini dimaksudkan untuk mengurangi risiko pemilihan pusat kluster. Kemudian dilanjutkan dengan metode *top-n* yang menggabungkan kluster terdekat dengan n

edge terpendek menyatu hingga jumlah kluster mencapai K . Dua strategi pemangkasan kluster yang meningkatkan efisiensi *K-Means* dengan menghapus kluster yang lebih jauh untuk mempercepat ruang pencarian, dapat disesuaikan untuk setiap iterasi. Kemudian memperluas percobaan untuk menentukan batas bawah pemangkasan jarak (k). Terakhir menggunakan teori pembaharuan yang didasarkan pada fakta bahwa jumlah titik yang bergerak di seluruh kluster akan berkurang secara signifikan seiring dengan kemajuan iterasi. Pertahankan nilai *K-Means* dengan poin yang diperbarui dan dipindahkan untuk meminimalkan biaya komputasi.

Pada penelitian [8] membahas tentang Algoritma yang umum digunakan untuk mengelompokkan data, seperti *K-Means*. Namun, metode ini memiliki beberapa kelemahan, seperti masalah akurasi data dari metode yang digunakan untuk mengukur kesamaan antara objek yang dibandingkan Untuk mengatasi permasalahan tersebut, penelitian ini membandingkan tiga metode (*Euclidean Distance*, *Manhattan Distance*, dan *Minkowski Distance*) untuk mengetahui situasi ketimpangan kebutuhan guru di kota Tegal. Dataset yang digunakan dalam penelitian ini adalah data dasar pendidikan dasar dan menengah di Kota Tegal. Hasilnya, akurasi dari ketiga metode yang dibandingkan adalah baik: 84,47% (*Euclidean Distance*), 83,85% (*Manhattan Distance*), dan 83,85% (*Minkowski Distance*). Dengan demikian, dapat disimpulkan bahwa metode *euclidean* merupakan metode terbaik untuk diterapkan dalam algoritma *K-Means Clustering*.

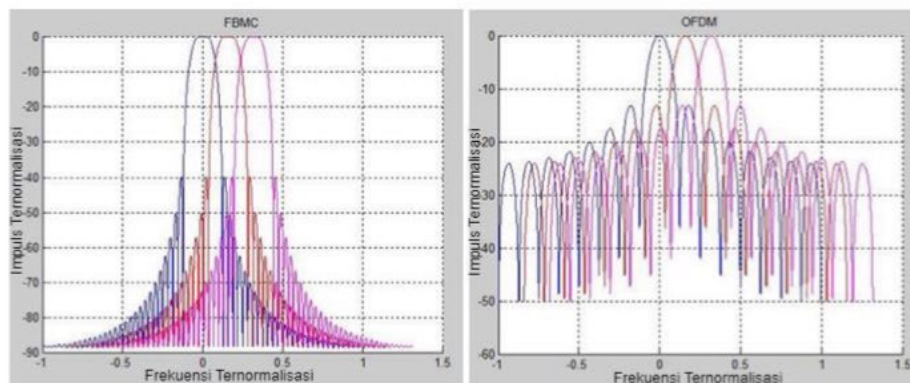
Pada penelitian [5] membahas tentang performansi teknik *multicarrier* FBMC-OQAM menggunakan *K-NN Clustering* sebagai pengganti *demapper*. Penelitian ini membandingkan sistem FBMC-OQAM dengan menggunakan algoritma pendeteksian simbol Zero Forcing (ZF) dan yang tidak menggunakan ZF. Modulasi yang digunakan untuk *demapper* adalah modulasi 16-QAM dengan parameter pengujian kinerja sistem yakni *Bit Error Rate* (BER), *Signal to Noise Ratio* (SNR), dan Kapasitas Kanal. Hasilnya menunjukkan sistem FBMC-OQAM dengan menggunakan dengan *KNN demapper* yang disertai penggunaan algoritma pendeteksi simbol *Zero Forcing* dapat meminimalisir nilai BER dibandingkan dengan sistem FBMC-OQAM tanpa menggunakan *Zero Forcing*. Hal tersebut dibuktikan oleh nilai BER pada FBMC-OQAM ZF dengan *K-NN Demapper*

mengalami penurunan nilai dari 0,4299 menjadi 0,002123 pada SNR 30 dB, dan untuk nilai BER pada sistem FBMC-OQAM memiliki nilai BER sebesar 0,5132 dan 0,5204 pada SNR 30 dB. Penggunaan SNR dengan nilai yang semakin besar menghasilkan kerapatan pada diagram konstelasi kaitannya dengan pengelompokan data yang semakin baik.

2.2 DASAR TEORI

2.2.1 Filter Bank Multi Carrier (FBMC)

Filter bank multi carrier (FBMC) merupakan jenis *multicarrier* dari hasil pengembangan dari sistem *multicarrier* OFDM dimana data yang dikirimkan dibagi pada *subcarrier* yang disusun secara *overlap* sehingga antar *subcarrier* akan saling ortogonal. FBMC dapat mengatasi beberapa kekurangan dari OFDM. Kekurangan dari OFDM dimana menggunakan *cyclic prefix* (CP) yang mengakibatkan pemborosan *bandwidth*. *Cyclic prefix* merupakan salinan bagian akhir dari sinyal informasi yang diletakan pada bagian awal sinyal informasi. Sedangkan pada FBMC memiliki penggunaan kapasitas kanal atau *bandwidth* yang jauh lebih baik dan mampu menawarkan kecepatan data yang lebih tinggi dalam spektrum *bandwidth* radio tertentu [9], seperti yang ditunjukkan pada Gambar 2.1. Berdasarkan Gambar 2.1 FBMC memiliki tingkat efisiensi spektrum yang lebih tinggi.

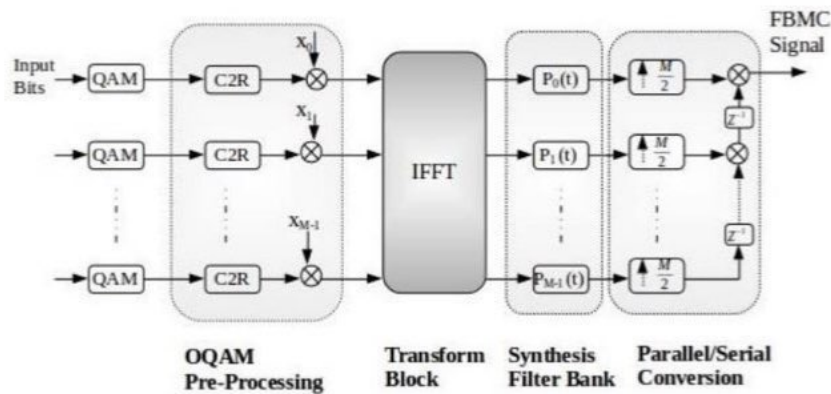


Gambar 2. 1 Perbandingan Spektrum Frekuensi FBMC dan OFDM [10].

Kelemahan lain dari OFDM adalah penguraian spektral yang lemah dari *subcarrier* sehingga mengakibatkan kobocoran spektral dan interferensi sinyal asinkron [9]. Sedangkan kelebihan dari FBMC sebagai berikut [11]:

1. Teknik *multicarrier* FBMC tidak menggunakan *cyclic prefix* atau *guard interval*.
2. Jika menggunakan modulasi OQAM, kapasitas dari *bandwidth* transmisi dapat terpenuhi.
3. *Sub-channel* pada FBMC dapat dikelompokkan ke dalam blok independen yang dibutuhkan untuk skalabilitas dan akses dinamis.
4. Apabila terjadi kendala frekuensi pada sinyal yang ditransmisikan dapat mudah diatasi.

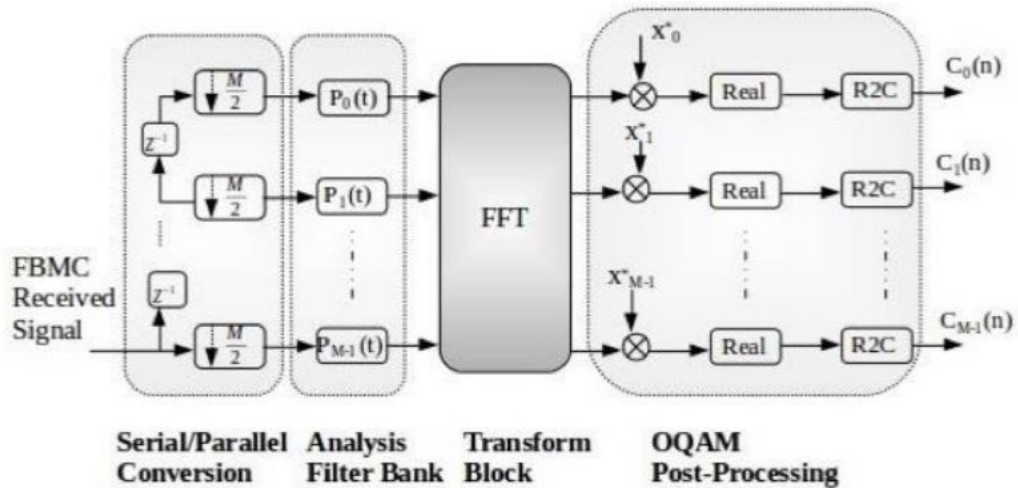
Sistem FBMC memiliki dua proses pentransmisian sinyal yaitu *pre-processing* dan *post-processing*. Proses *pre-processing* di FBMC disebut *synthesis filter bank* dan *post-processing* disebut *analysis filter bank*. *Synthesis filter bank* ditempatkan dalam proses transmisi data tepatnya setelah proses *pre-processing* OQAM. Sedangkan di sisi penerima (*receiver*), proses analisis terjadi sebelum proses *post-processing* OQAM [12].



Gambar 2. 2 Konfigurasi pada *Synthesis Filter Bank* [13].

Gambar 2.2 merupakan konfigurasi pada *Synthesis Filter Bank*. *Synthesis filter bank* terdiri dari dua proses yaitu proses IFFT dan proses filter. Sebelum masuk ke proses IFFT, output dari proses *preprocessing* QAM ($x_0[k]$ hingga $x_{M-1}[k]$) untuk setiap *subchannel* (0 hingga $M-1$, dimana M adalah jumlah *subchannel*) akan dinetralkan terlebih dahulu dengan dikalikan dengan β ($\beta=1$). *Synthesis Filter Bank* dalam filter, disimbolkan dengan $A(z^2)$, digunakan untuk memisahkan sinyal menurut frekuensi. Jenis filter yang digunakan dalam *synthesis filter bank* dapat menggunakan berbagai jenis filter yang berbeda. Jenis filter yang paling mudah adalah filter yang ideal. Konsep pada filter ideal adalah frekuensi yang ada akan langsung diteruskan atau diloloskan tanpa adanya peredaman. Keluaran dari

synthesis filter bank masing-masing akan dikalikan dengan peningkatan laju sebesar $M/2$ kali (upsampling $M/2$) dan diberikan penundaan masing-masing 1 kali yang disimbolkan dengan z^{-1} agar mendapatkan proses perubahan dari semula bentuk paralel menjadi bentuk seri dengan keluaran $s[n]$ [14].



Gambar 2. 3 Konfigurasi pada *Analysis Filter Bank* [13].

Gambar 2.3 merupakan konfigurasi pada *Analysis Filter Bank*. *Analysis filter bank* pada FBMC terdiri dari 2 proses yaitu proses FFT dan proses filter. Sebelum masuk ke proses FFT data masukan $r[n]$ yang berbentuk data seri akan diubah menjadi bentuk paralel dengan cara pemberian penundaan pengiriman data sebesar 1 kali (z^{-1}), kemudian diturunkan laju pengiriman datanya sebesar faktor 2 kali ($\downarrow 2$ atau *down sampling*. Keluaran dari proses sebelumnya, maka setiap *subchannel* dari *subchannel* 0 hingga *subchannel* $M-1$ difilter $A(z^2)$. Filter dalam *analysis filter bank* digunakan untuk menggabungkan sinyal berdasarkan frekuensi. Jenis filter yang digunakan dalam *analysis filter bank* harus sesuai dengan jenis filter yang digunakan dalam proses *synthesis filter bank*. Setelah dikenakan filter selanjutnya data keluaran filter yang masing berbentuk sinyal berdomain waktu diubah kembali menjadi domain frekuensi dengan melalui proses FFT, yang selanjutnya masing-masing sub saluran keluaran FFT dinetralkan dengan dikalikan β^* ($\beta^*=1$) sehingga menghasilkan keluaran $y_0[k]$ sampai $y_{M-1}[k]$ yang selanjutnya diproses ke dalam pasca pengolahan OQAM atau *OQAM Post Processing* [14].

2.2.2 Additive White Gaussian Noise (AWGN)

Additive White Gaussian Noise (AWGN) adalah *noise* yang ditambahkan ke kanal mengikuti distribusi *Gaussian*/normal yang terjadi pada semua frekuensi dengan kerapatan spektral daya konstan. *Noise* ini disebut *white* karena terdiri dari seluruh frekuensi dalam spektralnya sebagai cahaya putih. Ketika daya *noise* memiliki *spectral density* yang seragam untuk setiap frekuensi maka dapat disebut sebagai *white noise*. Pada sistem komunikasi *noise* AWGN mempunyai karakteristik yaitu spektrum daya *noise* dianggap memiliki harga yang sama di setiap frekuensi. Selain itu pola munculnya *noise* dianggap terdistribusi *gaussian* dengan nilai rata-rata nol dan varian tergantung rapat daya yang diperkirakan *noise* tersebut. Spektral daya sinyal AWGN dinyatakan dengan [15] :

$$S_W(f) = \frac{N_o}{2} \quad (2.1)$$

Probabilitas *gaussian* pada AWGN dapat dilihat pada persamaan [15] :

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (2.2)$$

dimana :

$$\sigma^2 = \frac{N_o}{2} \quad (2.3)$$

Keterangan :

S_W : Spektral daya sinyal AWGN

N_o : *Noise*

$f(x)$: Probabilitas kemunculan derau AWGN

σ : Standar deviasi

2.2.3 Equalizer

Proses *equalization* merupakan proses mencari respon agar kanal dapat diatasi. Pada proses estimasi kanal maka akan didapatkan matrik respon kanal. *Equalizer* sering digunakan dalam domain waktu maupun domain frekuensi dalam sistem komunikasi tradisional. *Equalizer* pada domain waktu, yaitu dalam sistem OFDM tradisional pemerataan atau ekualisasi tentunya sangat diperlukan. Karena *equalizer* ini digunakan sebagai penyeimbang dari karakteristik kanal. Pada bagian penerima, *equalizer* menghasilkan karakteristik yang berlawanan dari kanal untuk mengimbangi ISI akibat waktu yang bervariasi dari kanal *multi-path*. Namun

equalizer dalam domain frekuensi merupakan suatu teknik penting yang digunakan untuk mengurangi ICI pada estimasi kanal. Hal ini disebabkan karena pengaruh dari sisa frekuensi *offset* dan efek pergeseran dari *doppler* terutama pada *channel* yang berubah cepat. Jenis variasi ekualisasi yang akan digunakan pada penelitian ini adalah *Zero Forcing* (ZF).

2.3.1 Zero Forcing (ZF)

Zero Forcing (ZF) adalah jenis *equalizer* atau algoritma pendeteksian simbol yang paling mudah. Dalam penggunaan algoritma *zero forcing* ini cukup sederhana yaitu dengan H adalah sebuah matriks kanal dan W adalah matriks yang menunjukkan proses linier pada penerima, sehingga pada ZF harus memenuhi syarat [16] :

$$WH = 1 \quad (2.4)$$

Supaya tiap-tiap elemen yang diharapkan dari simbol data dapat dideteksi, maka dibutuhkan proses “memaksa” interferensi menjadi nol. Matriks W merupakan matriks kebalikan atau kebalikan semu (*pseudo invers*, PI) dari matriks kanal H , seperti ditunjukkan pada persamaan berikut [16]:

$$W_{ZF} = H^{-1} \text{ untuk } N_{RX} \quad (2.5)$$

$$W_{ZF} = (H^H H)^{-1} H^H \text{ untuk } N_{TX} \neq N_{RX} \quad (2.6)$$

Dengan H^H merupakan konjugat transpos matriks H . *Pseudo Invers* (PI) terjadi ketika jumlah antenna pengirim (N_{TX}) lebih kecil atau sama dengan jumlah antenna penerima (N_{RX}), sedangkan untuk N_{TX} yang lebih besar dari N_{RX} , maka $H^H H$ bernilai singular sehingga kebalikannya tidak terdefinisi.

2.2.4 Modulasi Digital

Modulasi digital adalah proses penumpangan sinyal informasi yang berbentuk digital ke dalam sinyal *carrier* (pembawa) yang berbentuk sinyal analog (sinusoidal). Modulasi digital juga dapat dikatakan simbol-simbol digital menjadi bentuk gelombang yang sesuai dengan karakteristik kanal yang akan dilewati baik berupa amplitudo, frekuensi, fasa yang mana ketiganya merupakan parameter kunci pada modulasi digital.

Modulasi digital terbagi atas 3 jenis yaitu: *Amplitude Shift Keying* (ASK) yang mengubah parameter amplitudo sinyal pembawa sesuai dengan perubahan sinyal informasi yang berupa sinyal digital, *Frequency Shift Keying* (FSK) yang mengubah parameter frekuensi sinyal pembawa sesuai dengan perubahan sinyal informasi, dan yang terakhir *Phase Shift Keying* (PSK) yang mengubah parameter fasa sinyal pembawa sesuai dengan perubahan sinyal informasi [17].

2.2.4.1 Modulasi *Quadrature Amplitude Modulation* (QAM)

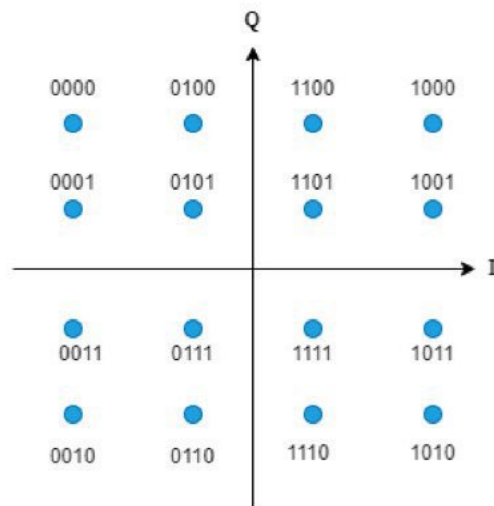
Modulasi QAM merupakan kombinasi atau gabungan antara modulasi digital ASK dan PSK. Hal tersebut menunjukkan pada modulasi QAM perubahan sinyal dipengaruhi oleh parameter amplitudo dan fasa. Modulasi ini membawa data dengan mengubah amplitudo dari dua sinyal pembawa yang berupa sinusoidal dan memiliki beda fasa 90^0 dari fasa lainnya oleh karena itu modulasi ini disebut dengan pembawa kuadratur. Modulasi QAM didefinisikan dengan persamaan [17]:

$$s(t) = I(t) \cdot \cos \omega_c t + Q(t) \cdot \sin \omega_c t \quad (2.7)$$

dimana $I(t) = A \cdot \cos \theta$ dan $Q(t) = A \cdot \sin \theta$.

Berdasarkan persamaan (2.7) di atas dapat ditemukan bahwa sinyal QAM terbentuk dari penjumlahan sebuah sinyal cosinus dengan amplitudo $I(t)$ dan sebuah sinyal sinus dengan amplitudo $Q(t)$. Amplitudo dan fasa pada masing masing simbol direpresentasikan dengan diagram dua dimensi yang disebut sebagai diagram konstelasi, seperti yang ditunjukkan pada Gambar 2.4.

Modulasi QAM memiliki beberapa bentuk, diantaranya: 8-QAM, 16-QAM, 64-QAM, dan 256-QAM. Pada penelitian ini menggunakan 16-QAM dimana untuk merepresentasikan satu buah simbol disusun oleh empat buah bit sinyal digital, yaitu 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1011, 1100, 1101, 1110, 1111.

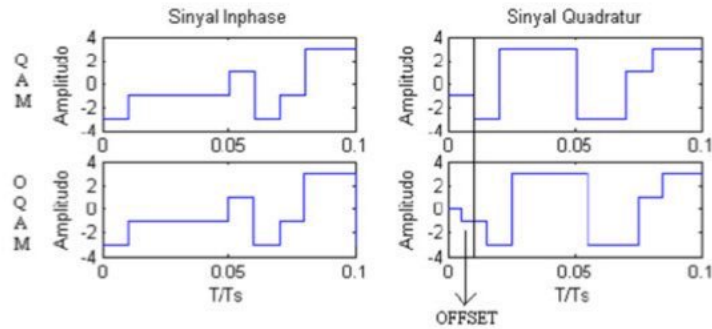


Gambar 2. 4 Diagram Konstelasi 16-QAM [17].

Berdasarkan Gambar 2.4 dapat diamati bahwa pada modulasi digital 16-QAM terdiri dari 16 simbol dimana tiap kuadran memiliki 4 simbol dan setiap simbol membawa 4-bit yang dibentuk dari sinyal I dan Q melalui *mapping* atau pemetaan menggunakan persamaan 2.7 [17].

2.2.4.2 Modulasi *Offset Quadrature Amplitude Modulation* (OQAM)

Modulasi digital QAM memiliki efisiensi dalam penggunaan *bandwidth* daripada modulasi lainnya. Namun QAM juga mempunyai kelemahan yakni *decision* yang kompleks serta *bit rate* yang rendah. Oleh karena itu penerapan modulasi QAM digantikan oleh modulasi *Offset QAM* yang saat ini telah dikembangkan dan memiliki kestabilan sistem yang lebih baik dalam pemakaian spektrum dan *bit error rate*. Implementasi modulasi OQAM dengan teknologi FBMC dapat meminimalisir terjadinya *Inter-Carrier Interference* (ICI) dan *Inter-Symbol Interference* (ISI) [11].



Gambar 2. 5 Modulasi Sinyal QAM dan *Offset*-QAM [9].

Gambar 2.5 menunjukkan perbedaan antara modulasi digital QAM dan *Offset* QAM. Berdasarkan gambar tersebut terlihat letak bit dalam kondisi sinyal *inphase* tetap pada koordinat aslinya sedangkan pada kuadratur terjadi pergeseran antar bit (memiliki *offset*). Pada skema *Offset* QAM, spektrum kanal yang berdampingan saling tumpang tindih tanpa menimbulkan *crosstalk* antar *subcarrier* yang diinduksi oleh *half time* simbol *delay* antara komponen *inphase* dan *quadrature* pada masing-masing *subcarrier*.

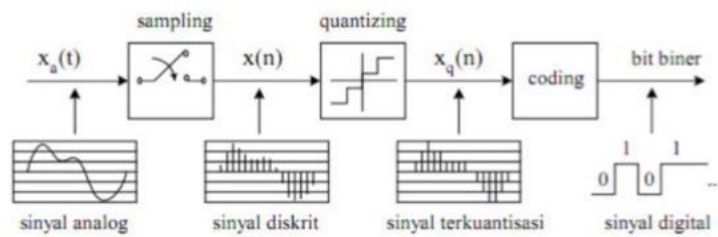
2.2.5 Sinyal Audio

Sinyal adalah sesuatu yang dapat diukur dan digunakan untuk memberikan informasi tentang fenomena fisik. Contoh sinyal yang dapat dijumpai dalam kehidupan sehari-hari, suara manusia, cahaya, suhu, kelembaban, gelombang radio, sinyal listrik, sinyal audio dan lain-lain. Sinyal audio secara khusus akan menjadi fokus pada penelitian ini sebagai proses akuisisi data. Secara matematis, sinyal biasanya dimodelkan sebagai fungsi dari beberapa variabel bebas. Contoh variabel bebas yang dapat digunakan untuk menggambarkan suatu sinyal adalah waktu, frekuensi, atau koordinat spasial.

Sinyal audio merupakan tipe sinyal analog atau sinyal waktu kontinyu. Oleh karena itu sinyal audio yang berbentuk analog tersebut perlu dilakukan proses konversi menjadi sinyal digital atau sinyal diskrit (bit). Terdapat dua faktor penting ketika mengkonversi sinyal analog menjadi sinyal digital. Yang pertama adalah *sampling rate*, atau seberapa sering nilai-nilai tegangan dicatat. Yang kedua adalah jumlah bit per sampel, atau keakuratan nilai yang direkam [18].

2.2.6 Konsep Pencuplikan Sinyal

Sinyal audio yang berbentuk analog perlu dilakukan proses perubahan sinyal analog menjadi sinyal digital sehingga sinyal tersebut dapat diolah pada komputer. Prosedur ini dinamakan konversi analog ke digital, dan sistemnya disebut *Analog to Digital Converter* (ADC). Secara konsep, konversi sinyal analog ke sinyal digital terdiri dari dua tahap seperti yang ditunjukkan pada Gambar 2.6 [19].



Gambar 2. 6 Konsep Pencuplikan Sinyal [19].

Berdasarkan Gambar 2.6 dapat diamati bahwa tahap pertama yaitu *sampling* (pencuplikan), dimana proses pengubahan sinyal waktu kontinu (analog) menjadi sinyal waktu diskrit (digital) diperoleh dengan mengambil cuplikan sinyal secara periodik, dengan periode cuplik T . Dengan kata lain, konversi sinyal analog waktu kontinu, $x_a(t)$ menjadi sinyal waktu diskrit bernilai kontinu, menghasilkan nilai $x(n)$ dengan mengambil cuplikan sinyal waktu kontinu pada saat waktu diskrit. Frekuensi *sampling* harus lebih besar dua kali dari frekuensi sinyal yang *disampling* dan harus memiliki gelombang yaitu puncak dan lembah. Proses *sampling* dapat ditulis secara matematis dengan [19]:

$$X(n) = x_a(nT) \quad (2.8)$$

Dengan T merupakan periode pencuplikan dalam detik dan n adalah bilangan bulat.

Kemudian tahap kedua yaitu *quantizing* (kuantisasi), dimana proses ini merupakan pengubahan sinyal dari sinyal waktu diskrit nilai kontinu menjadi sinyal digital (waktu diskrit bernilai diskrit). Dengan kata lain, $x(n)$ menjadi sinyal waktu diskrit bernilai diskrit $x_q(t)$. Nilai setiap waktu kontinu dikuantisasi atau dinilai dari tegangan pembanding yang terdekat. Selisih antara cuplikan $x(n)$ dan sinyal terkuantisasi $x_q(t)$ dinamakan *error* kuantisasi. Tegangan sinyal masukan (*input*) pada skala penuh dinyatakan dengan 2^N tingkatan. Dimana N adalah resolusi

bit ADC atau jumlah kedudukan tegangan pembanding yang ada. Tahap terakhir adalah pengkodean atau *coding* untuk setiap level tegangan pembanding kedalam barisan bit biner. Untuk $N = 4$ bit, maka level tegangan pembanding = 16 tingkatan. Keenam belas susunan tersebut dikodekan sebagai berikut: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, dan 1111 [19].

2.2.7 Clustering (Pengelompokan)

Clustering didasarkan pada informasi yang diperoleh dari data yang menggambarkan hubungan antar objek menggunakan prinsip untuk memaksimalkan kesamaan antara anggota kelas dan meminimalkan kesamaan antar kelas/*cluster* Proses pengelompokan atau pengkategorian data. *Clustering* membagi data ke dalam kelompok-kelompok yang berisi objek dengan karakteristik yang sama. Adapun tujuan dari *clustering*, yaitu untuk meminimalisasi variasi dalam satu *cluster* dan memaksimalkan variasi antar *cluster* [20].

2.2.7.1 Algoritma *K-Means*

K-Means merupakan jenis *clustering* dengan algoritma yang menggunakan parameter masukan dan memecah sekumpulan n objek ke dalam k *cluster* sehingga susunan antar anggota dalam satu *cluster* memiliki kemiripan yang tinggi sedangkan pada anggota *cluster* lain memiliki tingkat kemiripan yang rendah. Kemiripan anggota *cluster* diukur dengan kedekatan objek dengan nilai mean pada *cluster* atau dinamakan sebagai *centroid cluster* [20].

Istilah - istilah dalam *K-Means* [20]:

1. n data: data set yang akan diproses sejumlah n data dimana n data tersebut terdiri dari simbol – simbolnya.
2. K *centroid*: Inisialisasi dari sentral *cluster* data diproses sebanyak k dimana sentral awal tersebut digunakan sebanyak jumlah kelas yang akan dihasilkan. *Centroid* diperoleh secara acak dari n data set yang ada.

Adapun algoritma dasar yang dilakukan pada data *clustering* menggunakan metode *K-Means*, yaitu [21]:

1. Tentukan jumlah *cluster* dan sesuaikan dengan modulasi *M-ary* yang digunakan. Misalnya 4 untuk QPSK dan 2 untuk BPSK.

2. Tentukan *centroid* secara acak.
3. Hitung *centeroid* dari data dalam setiap *cluster* dengan menghitung rata-rata simbol data pada diagram konstelasi.
4. Alokasikan tiap – tiap data ke *centroid* dengan menghitung jarak terdekat. Pada penelitian ini menggunakan teknik *manhattan distance*.
5. Kembali ke step 3 jika masih ada data yang berpindah *cluster* atau jika terjadi perubahan nilai *centroid* ataupun nilai pada *objective* diatas nilai *threshold* yang telah ditetapkan. Proses ini akan terus berulang hingga *centroid* stabil (algoritma konvergen).

Jarak antar *centroid* dihitung menggunakan teknik *manhattan distance* atau dapat disebut juga dengan jarak *manhattan*. *Manhattan distance* digunakan untuk menghitung perbedaan absolut (mutlak) antara koordinat sepasang objek. Teknik perhitungan jarak *manhattan* dinyatakan dengan [22]:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2.9)$$

Keterangan :

- d = jarak antara x dan y
- x = data pusat klaster
- y = data pada atribut
- i = setiap data
- n = jumlah data
- x_i = data pada pusat klaster ke i
- y_i = data pada setiap data ke i

Manhattan distance mempunyai kelebihan dalam perhitungan metode *K-Means*. Adapun kelebihan dari *manhattan distance* adalah sebagai berikut [22]:

1. Rata-rata kesalahan kuadrat diantara nilai aktual dan nilai peramalan (*squared error*) menggunakan *bank dataset* dan diuji menggunakan tool WEKA.
2. *Manhattan Distance* cenderung lebih cepat pada perhitungan manual karena jumlah iterasi pada *manhattan distance* cenderung lebih sedikit dalam perhitungan manual.

2.2.7.2 Silhouette Coefficient

Metode *Silhouette Coefficient* merupakan metode evaluasi *cluster* yang menggabungkan metode *cohesion* dan *separation*. *Cohesion* diukur dengan menghitung semua objek dalam sebuah *cluster* dan *separation* diukur dengan menghitung jarak rata-rata setiap objek dalam suatu *cluster* dengan *cluster* terdekatnya. Berikut persamaan untuk menghitung *silhouette coefficient* [23]:

$$s_{(i)} = \frac{b_{(i)} - a_{(i)}}{\max(a_{(i)}, b_{(i)})} \quad (2.10)$$

$$a_{(i)} = \frac{1}{|A|-1} \sum_{j \neq i}^{\in A, j} J d(i, j) \quad (2.11)$$

$$d_{(i, C)} = \frac{1}{|A|} \sum_{j \in C} d_{i, j} \quad (2.12)$$

$$b_{(i)} = \min d(i, C), \quad C \neq A \quad (2.13)$$

Keterangan :

J = dokumen lain dalam cluster A

$D_{(i, j)}$ = jarak antar dokumen i dan j

$D_{(i, C)}$ = dari jarak rata-rata data i dengan semua dokumen di cluster lain C dengan syarat $A \neq C$

$a_{(i)}$ = rata rata jarak dari sebuah data dengan semua data dalam satu cluster

$b_{(i)}$ = nilai terkecil

2.2.8 Parameter Pengujian

Adapun parameter yang berperan dalam pengujian performansi teknik *clustering K-Means* sebagai fungsi *demapper* yaitu *bit error rate* (BER), *signal to noise ratio* (SNR), dan kapasitas kanal.

2.2.8.1 Bit Error Rate (BER)

Bit Error Rate (BER) atau yang biasa disingkat BER adalah salah satu parameter untuk menguji performansi dari suatu sistem komunikasi. *Bit Error Rate* merupakan perbandingan antara bit yang *error* dengan bit yang ditransmisikan secara keseluruhan. BER menjadi ukuran kredibilitas sinyal berdasarkan kapasitas atau presentase bit yang ditransmisikan dan diterima dalam kesalahan (kondisi *error*). Pada dasarnya, semakin banyak bit yang buruk maka semakin besar

dampaknya pada kualitas sinyal. *Bit Error Rate* didefinisikan dengan persamaan [17]:

$$BER = \frac{\text{Error Bit}}{\text{Transmitted Bit}} \quad (2.14)$$

Bit yang diterima dalam kondisi *error* dapat terjadi pada saat pengiriman (transmisi) data berlangsung. Hal tersebut disebabkan oleh beberapa faktor diantaranya gangguan derau (*noise*), interferensi, distorsi atau kesalahan sinkronisasi bit. Ketika jumlah bit yang *error* pada saat pengiriman semakin minim maka BER yang akan diperoleh juga semakin kecil sehingga kinerja sistem bisa dikatakan baik [17].

2.2.8.2 *Signal to Noise Ratio* (SNR)

Pada sistem transmisi sinyal, *Signal to Noise Ratio* (SNR) juga merupakan salah satu parameter yang harus diperhatikan. SNR adalah parameter yang digunakan untuk menampilkan jumlah derau (*noise*) yang mengganggu sinyal yang ditransmisikan. Dapat diartikan dengan kata lain SNR membandingkan daya yang diinginkan pada suatu sinyal terhadap daya yang terkandung oleh *noise* yang terjadi pada titik tertentu selama proses transmisi. Hubungan daya sinyal dengan *noise* didefinisikan sebagai berikut [24]:

$$SNR = 10 \log_{10} \frac{S}{N} \quad (2.15)$$

dengan:

SNR = *Signal to Noise Ratio* (dB)

S = Daya sinyal rata-rata (Watt)

N = daya derau/*noise* (Watt)

Keterkaitan antara *noise* dan SNR berbanding terbalik. Ketika *noise* memiliki nilai yang besar maka akan mengakibatkan nilai SNR yang semakin kecil dan sebaliknya. Hal tersebut menunjukkan kualitas sinyal akan baik ketika nilai SNR lebih besar karena *noise* yang dihasilkan kecil. Dan juga ketika jarak transmisi dekat, maka kekuatan SNR akan semakin besar begitupun sebaliknya [24].

2.2.8.3 Kapasitas Kanal

Kapasitas kanal merupakan salah satu parameter untuk mengetahui performa dari kanal nirkabel. Kapasitas kanal adalah banyaknya informasi yang

dapat dibawa oleh sebuah saluran transmisi dengan satuan bit per detik. Nilai kapasitas kanal dipengaruhi oleh *bandwidth* dan *Signal to Noise Ratio* (SNR), dengan persamaan berdasarkan dengan teorema *Shannon-Hartley* adalah sebagai berikut [25]:

$$C = \log_2 (1 + SNR) \quad (2.16)$$

Dengan:

$C = \text{Capacity}$ (bps/hz)

$SNR = \text{Signal to Noise Ratio}$ (dB)

Selama proses transmisi kanal nirkabel, *noise* AWGN menjadi pengusik yang tidak dapat dihindari. Oleh karena itu penentuan batas pesat bit penting untuk mengetahui jumlah maksimum informasi yang dapat terkirim dari *transmitter* sampai *receiver*. Semakin besar nilai kapasitas kanal dari suatu sistem maka akan semakin baik pula kinerja sistem tersebut [25].