

## BAB 2

### DASAR TEORI

#### 2.1 KAJIAN PUSTAKA

Pada penelitian yang dilakukan oleh Alireza Borhani dan Matthias Patzold pada tahun 2012 yang berjudul *Modeling of Vehicle-to-Vehicle Channels in the Presence of Moving Scatterers* membahas tentang pemodelan kanal *Vehicle to Vehicle* dengan asumsi skenario propagasi dimana *scatterers* lokal bergerak dengan kecepatan acak dan arah acak. Dalam komunikasi V2V mempunyai kecepatan yang tinggi sedangkan *mobile to mobile* mempunyai kecepatan yang rendah. Pada penelitian ini V2V mempunyai tiga parameter yaitu pergerakan disisi *Transmitter* (Tx), pergerakan disisi *Receiver* (Rx) dan Pergerakan disisi *Scatter*. Dari ketiga pergerakan tersebut timbul yang namanya efek *Doppler* yang mengakibatkan antara pengirim dan penerima tidak bisa berkomunikasi. Pada penelitian ini menggunakan *Auto Correlation* (ACF) sebagai fungsi validasinya [7].

Penelitian yang dilakukan oleh Parnika Kansal, Ashok Kumar Shankhwar, pada tahun 2017 yang berjudul “*FBMC vs OFDM Wafeform Contenders for 5G Wireless Communication System*”. Pada penelitian ini membahas mengenai perbandingan kinerja OFDM dan FBMC sebagai pesaing paling potensial dari jaringan 5G menggunakan simulasi MATLAB. Hasil penelitian ini menunjukkan bahwa teknik FBMC merupakan solusi yang tepat untuk komunikasi nirkabel masa depan khususnya pada jaringan 5G karena *filter* yang digunakan oleh FBMC dirancang dengan prinsip *Nyquist* yang dapat mengurangi masalah kebocoran *spektral* OFDM dan menghasilkan ICI atau ISI yang dapat diabaikan [11].

Penelitian yang dilakukan oleh Satwinder Kaur, Lavish Kansal, Gurjot Singh Gaba, dan Nuru Savarof yang berjudul “*Survey of Filter Bank Multicarrier (FBMC) as an efficient waveform for 5G*”. Penelitian ini bersifat survei terkait metode baru pada sistem multicarrier yang akan digunakan sebagai pengembangan metode OFDM dan akan diterapkan pada jaringan 5G. Hasil dari survei ini adalah FBMC merupakan modifikasi dari OFDM. Dalam OFDM *cycle prefix* (CP) digunakan sebagai ketahanan sinyal. CP adalah teknik modulasi untuk mengatasi *Inter Symbol Interference* (ISI) dan *Inter Carrier Interference* (ICI). Salah satu masalah yang

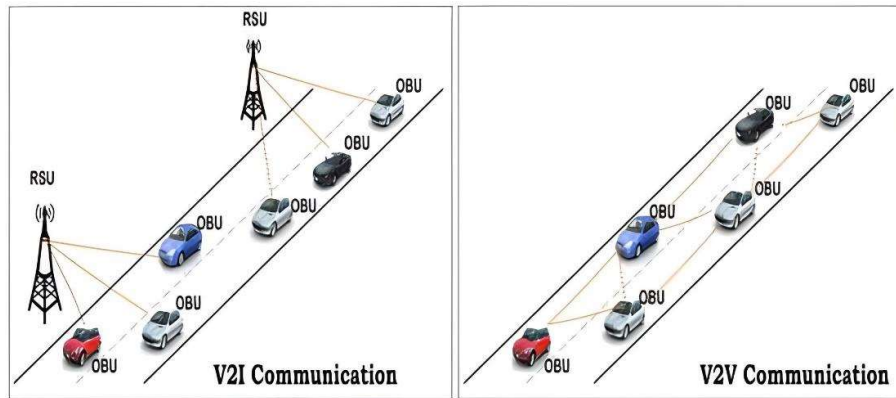
paling signifikan dalam sistem jaringan adalah gangguan antar simbol. Kemampuan teknik modulasi bergantung pada strategi yang digunakan untuk memodulasi sinyal dan mengurangi level *bit error* dan derau antar simbol. Metode modulasi yang berbeda penggunaannya untuk *multi carrier* FBMC. Dari jumlah tersebut, beberapa metode lebih andal serta mendapatkan sinyal *output* yang lebih baik [12].

Dalam penelitian yang dilakukan oleh Rizky Wahyudi, Arfianto Fahmi, Aief Diaz Pambudi pada tahun 2016 yang berjudul “Analisis Penanggulangan *Inter Carrier Interference* (ICI) di OFDM menggunakan *Zero Forcing Equalizer*”. Penelitian ini membahas metode penanggulangan ICI menggunakan estimasi matriks kanal dan ekualisasi *Zero Forcing* (ZF). Estimasi matriks kanal dilakukan dengan estimasi nilai *Carrier Frequency Offset* (CFO) setelah dibuat invers matriks kanal yang akan menjadi respon *equalizer*. Proses ekualisasi dapat mengembalikan penempatan simbol modulasi yang tersebar akibat ICI. Ekualisasi dilakukan dengan mengkalikan respon kanal dan simbol *output* FFT. Dari hasil penelitian tersebut, dapat diketahui bahwa *Inter Carrier Interference* (ICI) berpengaruh besar dalam performansi OFDM, tetapi dengan penambahan *Zero Forcing Equalizer* ICI dapat diatasi dibuktikan dengan peningkatan performansi sebesar 2 dB [13].

## **2.2 DASAR TEORI**

### **2.2.1 Sistem Komunikasi *Vehicle to Vehicle***

Sistem komunikasi *Vehicle to Vehicle* (V2V) merupakan sistem komunikasi yang memungkinkan kendaraan berkomunikasi dengan kendaraan lain, kendaraan dengan infrastruktur dan kendaraan lingkungan sekitarnya. Sejak komunikasi nirkabel dari *Intelligent Transportation System* (ITS) berkembang dengan pesat, maka diperlukan berbagai layanan yang dapat menjamin hubungan komunikasi antar kendaraan, kendaraan dengan infrastruktur dan kendaraan dengan lingkungan sekitar [14].



**Gambar 2.1 Komunikasi V2V dan V2I [15]**

Pada Gambar 2.1 merupakan Pengoperasian sejumlah protocol WAVE telah memungkinkan terjadinya komunikasi pertukaran informasi antara kendaraan dengan infrastruktur atau disebut *Vehicle to Infrastructure* (V2I) dan pertukan informasi antar kendaraan atau disebut *Vehicle to Vehicle* (V2V). Perpaduan kedua jenis komunikasi ini ditetapkan sebagai dasar sistem komunikasi pada teknologi VANET dimana sejumlah kendaraan dapat berkomunikasi menggunakan jaringan nirkabel baik secara infrastruktur maupun ad-hoc. Standar WAVE dibangun di atas dua unit dasar, yaitu disebut RSU (*Road Side Unit*) dan OBU (*Onboard Unit*). RSU adalah unit infrastruktur yang merupakan jalur akses (*Access Point*) untuk menghubungkan pertukaran komunikasi pada kendaraan sehingga dapat mengakses jaringan. Sedangkan OBU adalah perangkat jaringan yang berada pada kendaraan. Sebuah OBU dapat berkomunikasi dan mengubah pesan ke OBU lain. Pertukaran informasi antar OBU disebut sistem komunikasi V2V. Sementara pertukaran informasi antara OBU dengan RSU disebut sistem komunikasi V2I [15].

### **2.2.2 Modulasi Digital**

Dalam modulasi digital sinyal informasi ditumpangkan pada sinyal pembawa. Sinyal informasi dapat ditumpangkan dengan mengubah amplitudo, frekuensi atau *fase* dari sinyal pembawa. Untuk mentransmisikan pesan berbentuk digital, pada sistem modulasi digital memberikan slot *time* ke sinyal dan mendapatkan fungsi kontinu yang mewakili simbol. Modulasi digital mencakup tiga Teknik modulasi dasar termasuk modulasi sinyal pembawa yang disebut ASK (*Amplitude Shift*

*Keying*), modulasi digital dengan mengubah frekuensi sinyal pembawa disebut FSK (*Frequency Shift Keying*), dan modulasi digital dengan mengubah fasa sinyal pembawa yang disebut dengan PSK (*Phase Shift Keying*). Adapun bentuk variasi modulasi digital lainnya adalah *Quadrature Amplitude Modulation* (QAM), yaitu suatu bentuk modulasi yang mengubah amplitudo dan *fase* pembawa [16].

### 2.2.3 Modulasi QAM

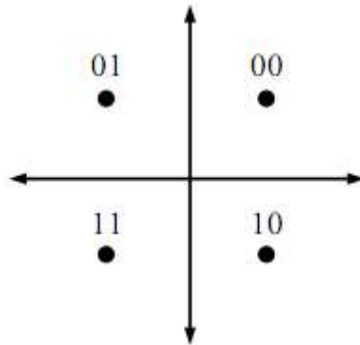
*Quadrature Amplitude Modulation* (QAM) adalah metode transmisi pada kecepatan *bit* yang lebih tinggi pada satu atau lebih saluran dengan *bandwidth* yang terbatas. Informasi yang dikirimkan dalam bentuk QAM terlebih dahulu dilakukan perubahan menjadi sebuah simbol QAM yang diwakilkan sebagai sinyal analog modulasi. Sinyal modulasi ini merubah *amplitude* dan *fase* sinyal pembawa mewakili simbol QAM, yang terdiri dari beberapa bit informasi.

Ada beberapa jenis modulasi dalam modulasi QAM, yakni 4-QAM, 16-QAM dan 64-QAM. Antara ke tiga jenis modulasi tersebut terdapat perbedaan pada jumlah *output* serta pada tiap bit yang mewakilinya. Modulasi 4-QAM memiliki empat tahap *output* yang berbeda yang terdiri dari 2 bit. Modulasi 16-QAM memiliki enam belas *output* dengan perbedaan amplitudo dan *fase*. Sinyal digital 4-bit diperlukan untuk mewakili simbol 16-QAM dan modulasi 16-QAM memerlukan simbol yang berbeda, masing-masing simbol terdiri dari 6 bit. Ada beberapa rumus standar 3GPP untuk modulasi QAM sebagai berikut.

Untuk menentukan nilai real dan imajiner pada Modulasi 4 QAM menggunakan persamaan 2.1 dibawah ini [17]:

$$x = \frac{1}{\sqrt{2}} \left\{ \left[ (1 - 2b(i)) + j(1 - 2b(i+1)) \right] \right\} \quad (2.1)$$

Dengan  $b(i)$  = bit pertama,  $b(i+1)$  = bit kedua.



**Gambar 2.2 Diagram Konstelasi 4 QAM[17]**

Untuk menentukan nilai real dan imajiner pada Modulasi 16-QAM menggunakan persamaan 2.2 dibawah ini untuk dapat dilakukan pemetaan (*mapping*) 16-QAM [17] :

$$x = \frac{1}{\sqrt{10}} \left\{ (1-2b(i)) [1-2b(i+2)] + j(1-2b(i+1)) [2-(1-2b(i+3))] \right\} \quad (2.2)$$

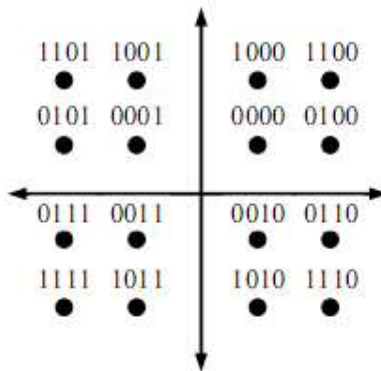
Keterangan :

b(i) = *bit* pertama

b(i+1) = *bit* kedua

b(i+2) = *bit* ketiga

b(i+3) = *bit* keempat



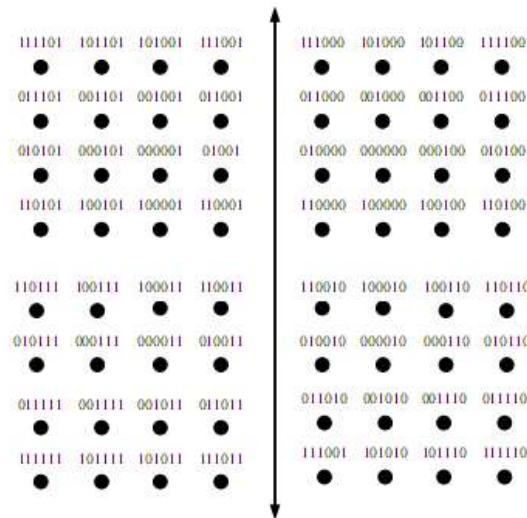
**Gambar 2.3 Diagram Konstelasi 16 QAM[17]**

Modulasi 16-QAM dapat digambarkan dalam sebuah diagram konstelasi seperti pada Gambar 2.3. Pada diagram konstelasi 16-QAM di mana masing-masing titik mewakili 4 bit simbol yang terbentuk melalui sinyal I dan Q dengan proses pemetaan (*mapping*).

Untuk menentukan nilai real dan imajiner pada Modulasi 64-QAM menggunakan persamaan 2.3 dibawah ini untuk dapat dilakukan pemetaan (*mapping*) 64-QAM [17] :

$$x = \frac{1}{\sqrt{42}} \left\{ \begin{array}{l} (1-2b(i))[4-(1-2b(i+2))[2-(1-2b(i+4))]] \\ +j(1-2b(i+1))[4-(1-2b(i+3))[2-(1-2b(i+5))]] \end{array} \right\} \quad (2.3)$$

Keterangan untuk  $b(i) = \text{bit pertama}$ ,  $b(i+1) = \text{bit kedua}$ ,  $b(i+2) = \text{bit ketiga}$ ,  $b(i+3) = \text{bit keempat}$ ,  $b(i+4) = \text{bit kelima}$ ,  $b(i+5) = \text{keenam}$ .

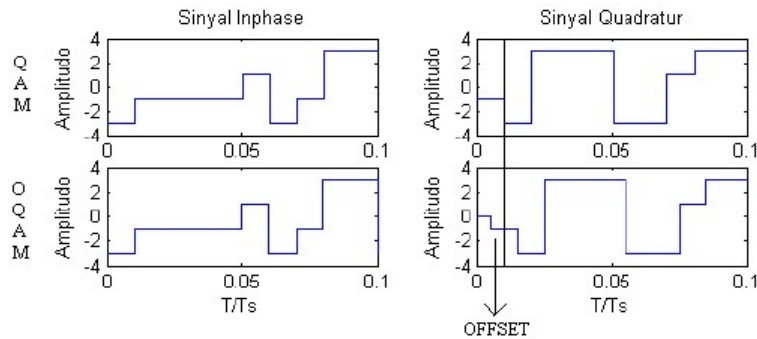


**Gambar 2.4 Diagram Konstelasi 64 QAM[17]**

Modulasi 64-QAM dapat digambarkan pada Gambar 2.4 yang merupakan sebuah diagram konstelasi. Pada diagram konstelasi 64-QAM di mana masing-masing titik mewakili 6 bit simbol yang terbentuk melalui sinyal I dan Q dengan proses pemetaan (*mapping*).

## 2.2.4 Modulasi Offset QAM

Modulasi yang biasa dipergunakan selama transmisi adalah QAM dengan *guard interval* dan *offset* QAM (OQAM) dengan *pulse shaping*. Sistem OQAM berbentuk pulsa mempunyai stabilitas sistem OQAM lebih baik dibandingkan dengan QAM konvensional. Dalam OQAM, spektrum saluran yang berdekatan tumpang tindih tanpa gangguan antar *subcarrier* karena penundaan setengah simbol waktu antara komponen *inphase* dan *quadrature* sinyal pada setiap *subcarrier*.



**Gambar 2.5 Sinyal modulasi pada QAM dan OQAM[18]**

Gambar 2.5 menunjukkan perbedaan antara QAM dan OQAM, bit-bit pada posisi *inphase* tetap sama, sedangkan terjadi pergeseran bit pada posisi *quadrature*. Pada modulasi OQAM terjadi pergeseran fasa hingga  $0^\circ$  dan  $\neq 90^\circ$  setiap  $T$  detik, tidak seperti QAM yang terjadi lompatan fasa hingga  $180^\circ$ . Pada setiap kanal bagian *riil* dan *imajiner* disimbolkan dengan, kemudian di *filter* menggunakan *pulse shaping*  $h(m)$  dan  $h(m+1)$ . Kedua bagian kedua tersebut dilakukan penjumlahan dan digeser pada frekuensi yang telah ditentukan dengan menggunakan modulasi *baseband*. Berikut merupakan *output* modulator [18]:

$$x(m) = \sum_{k=0} \sum_{n=0}^{N-1} [a_{k,n} h(m-kN) + jb_{k,n} h(m-kN + \frac{N}{2})] e^{j(\frac{2\pi}{N}m + \frac{\pi}{2})n} \quad (2.4)$$

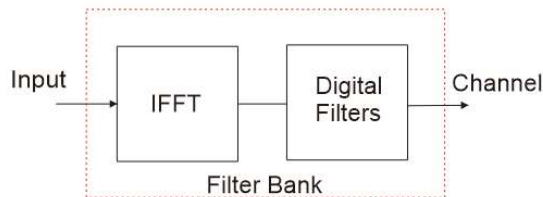
Dari persamaan diatas didapatkan persamaan [2.5]:

$$C_{k,n} = a_{k,n} + jb_{k,n} \quad (2.5)$$

Dimana  $C_{k,n}$  adalah simbol bagian *real* dan *imajiner*.

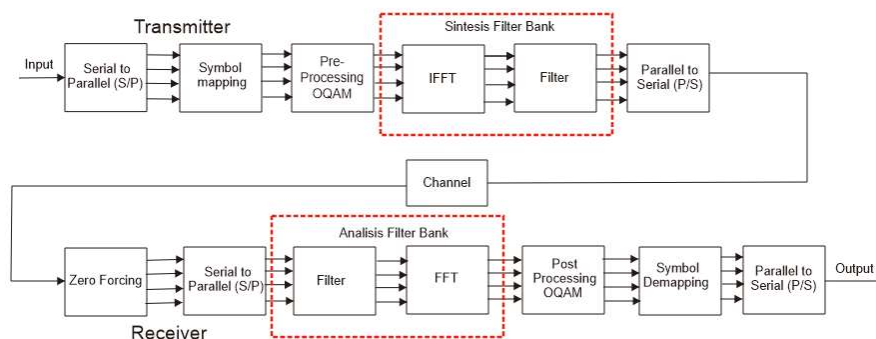
### 2.2.5 Filter Bank Multi Carrier (FBMC)

*Filter Bank Multi Carrier/ Offset Quadrature Amplitude Modulation* (FBMC/OQAM) merupakan perkembangan dari *Orthogonal Frequency Division Multiplexing* (OFDM) yang bertujuan untuk melengkapi kekurangan yang dimiliki OFDM. Adapun kekurangan OFDM yaitu besarnya nilai *Peak to Average Power Ratio* (PAPR), sensitif terhadap *Carrier Frequency Offset*, serta sulitnya mengoperasikan *Fast Fourier Transform* (FFT). Adanya *Carrier Frequency Offset* membuat penerima akan sulit mendeteksi sinyal OFDM serta menyebabkan *Frequency shifting* yang akan membuat hilangnya orthogonalitas sinyal OFDM. Dengan adanya FBMC sehingga dimodifikasi dengan menggunakan filter untuk mengurangi *noise*.



**Gambar 2.6 Basic Diagram *Filter Bank* [12]**

Dapat dilihat pada Gambar 2.6 pada FBMC memakai (IFFT) tapi tidak memakai (CP), sebagai gantinya dipakailah *filter*. Pada FBMC modulasi yang dipakai menggunakan OQAM. Dibawah ini adalah Gambar 2.7 merupakan blok diagram FBMC.



**Gambar 2.7 Blok Diagram FBMC [11]**

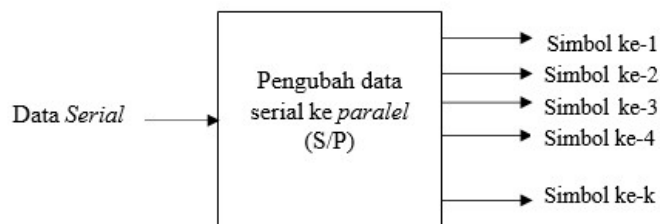


Pada Gambar 2.7 di sisi *transmitter* digunakan IFFT sebagai modulator. Terdapat beberapa bagian di sisi *transmitter* diantaranya adalah:

### 2.2.5.1 Blok Sisi *Transmitter*

#### a. Pengubahan Data Dari Serial to Paralel (S/P)

Blok Serial ke paralel ditunjukkan pada Gambar 2.8 berfungsi untuk mengubah format data serial ke paralel dengan Langkah mengelompokkan data sesuai berdasarkan bentuk matriks bit, dimana jumlah baris menunjukkan jumlah *subcarrier* yang digunakan.



**Gambar 2.8 Pengubah Data Serial ke Paralel**

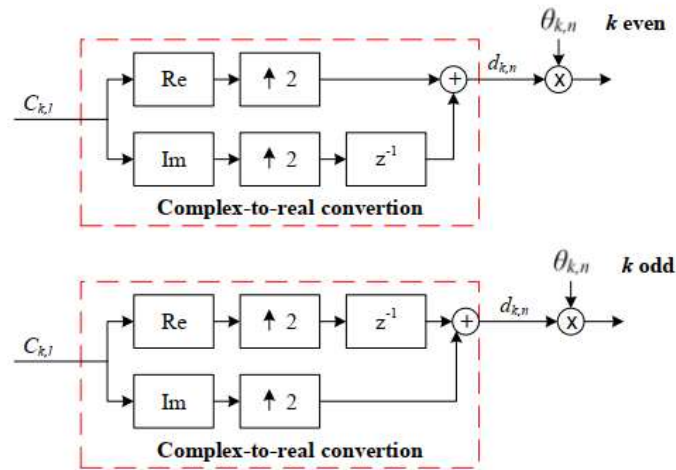
#### b. Simbol *Mapping*

Peta simbol modulasi digunakan untuk menghasilkan sinyal listrik termodulasi 16QAM, dan kemudian simbol modulasi *demapper* mendemodulasi sinyal sesuai dengan jenis modulasi yang digunakan. Jenis simbol *mapper* harus sesuai dengan jenis modulasi sinyal asli yang ditransmisikan harus sesuai.

#### c. Pra Pengolahan OQAM

Simbol yang dikirim ke proses pra pengolahan OQAM dibagi menjadi 2 simbol ganjil dan genap. Proses pengolahan simbol terdiri dari dua proses yaitu proses pengubahan bilangan kompleks menjadi bilangan *riil* dan proses perkalian dengan  $\theta_{kn}$ . Pada proses pertama bertujuan untuk membuat simbol baru. Simbol ganjil dan genap masing-masing akan dipisah menjadi bilangan *riil* dan kompleks, Masing-masing bilangan *riil* dan kompleks akan mengalami peningkatan jumlah (*upsampling* sebesar 2 kali) dan bergeser setengah fasa sebesar  $90^\circ$  (1 fasa =  $180^\circ$ ). Perbedaan di dalam proses pengolahan terletak pada tahap pergeseran fasa. Simbol genap terjadi pergeseran fasa pada bilangan imajiner. Sedangkan simbol ganjil

terjadi pergeseran fasa pada bilangan riil. Proses pengolahan dijelaskan pada Gambar 2.9.



**Gambar 2.9** Proses Pengolahan OQAM Untuk Saluran ganjil dan genap

Pada proses kedua simbol yang dihasilkan oleh proses pertama dikalikan dengan  $\theta_{kn}$  untuk mendapatkan nilai riilnya dengan persamaan (2.6) [11] :

$$\theta_{kn} = j^{(k+n)} \quad (2.6)$$

Dengan  $j$  adalah simbol imajiner,  $k$  merupakan urutan kanal, dan  $n$  adalah jumlah kanal. Pada simbol genap dan ganjil terdapat pergeseran setengah fasa. Simbol genap memiliki pergeseran setengah fasa pada bilangan imajiner, sedangkan simbol ganjil terjadi pergeseran setengah fasa pada bilangan riil.

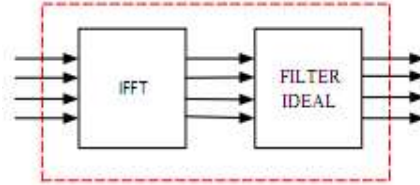
*d. Sintesis Bank Filter*

Pada proses sintesis *Bank Filter* terdiri dari 2 proses yaitu proses kebalikan dari transformasi *fourier* dan proses *filter* atau *bank filter*. *Filter* yang digunakan menggunakan *filter* ideal. Untuk mendapatkan data yang diterima, *filter* ideal memiliki persamaan (2.7) [11]:

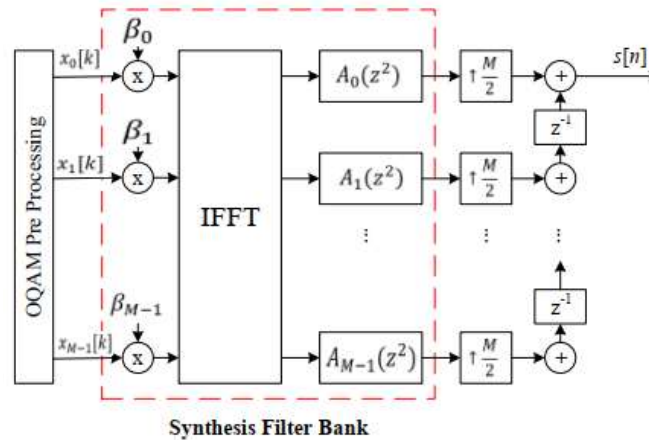
$$y = h(x) + n \quad (2.7)$$

Dimana  $y$  adalah data yang diterima,  $hx$  adalah data yang dikirim dikalikan dengan *channel*, dan  $n$  adalah *noise*. Pada *Filter* ini diasumsikan bahwa *channel* sama dengan satu artinya bahwa berapa pun data yang masuk akan ditransmisikan dengan

pengalinya sama dengan satu. Sehingga simbol yang akan ditransmisikan langsung diteruskan tanpa adanya data yang dihilangkan. Proses *synthesis bank filter* seperti ditunjukkan pada Gambar 2.10.



Gambar 2.10 Proses *Syntesis Bank Filter* [10]

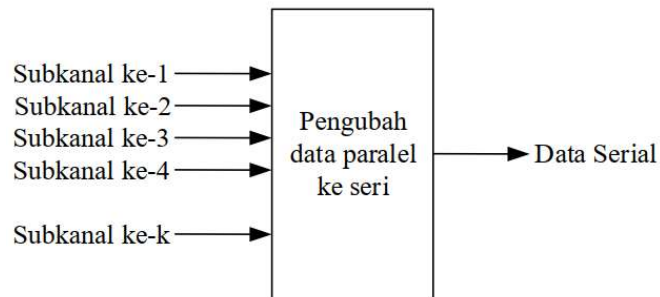


Gambar 2.11 Konfigurasi pada *Syntesis Bank Filter* [10]

Sesuai dengan Gambar 2.11 *Syntesis filter bank* mempunyai 2 proses yaitu proses IFFT serta proses *filter*. *Filter* di *synthesis bank filter* digunakan untuk membedakan sinyal berdasarkan frekuensinya. Jenis yang digunakan pada *synthesis filter bank* dapat menggunakan berbagai jenis *filter*. Jenis *filter* yang paling sederhana adalah *filter* ideal. *Filter* ideal adalah frekuensi yang ada diteruskan atau ditransmisikan secara langsung tanpa redaman[10].

e. Pengubahan Data Paralel ke Seri (P/S)

Pada proses pengubahan ke seri, setiap data diterima dalam waktu yang berbeda, bukan hanya pada satu waktu. Proses ini di sisi pengirim berfungsi untuk mengubah data dari paralel ke seri pada keluaran *synthesis bank filter* agar dapat dikirimkan menjadi 1 baris. Proses pengubahan data paralel ke seri ditunjukkan pada Gambar 2.12.



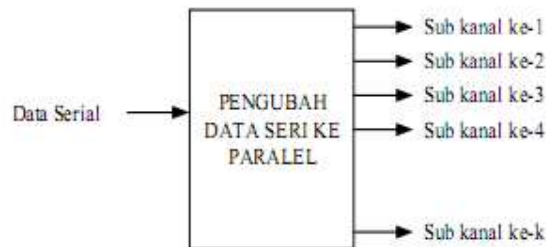
**Gambar 2.12** Pengubahan Data Paralel ke Serial

#### 2.2.5.2 Blok Sisi Receiver

Pada Gambar 2.7 di sisi *receiver* digunakan FFT sebagai *demodulator*. Terdapat beberapa bagian di sisi *receiver* di antaranya adalah:

- a. Pengubah Serial menjadi Paralel (S/P)

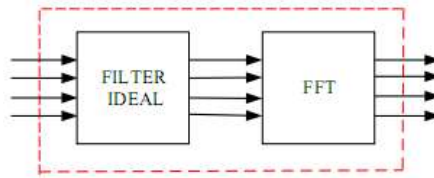
Proses pengubahan data serial menjadi paralel adalah dengan mengubah bentuk bentuk keluaran antena *receiver* menjadi bentuk paralel kemudian dilanjutkan ke proses berikutnya yaitu proses *analysis bank filter*. Perubahan data paralel menjadi seri diperlihatkan pada Gambar 2.13.



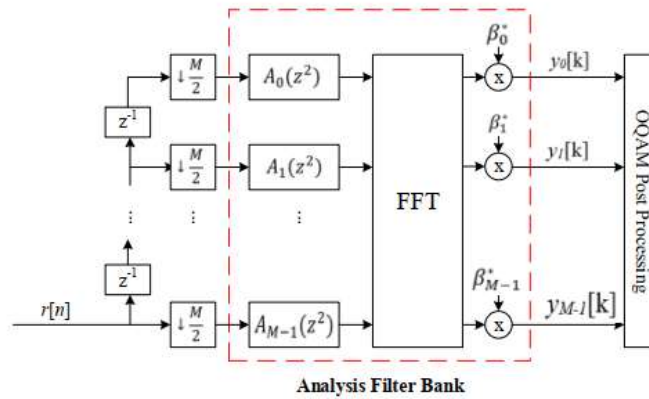
**Gambar 2.13** Pengubah data paralel ke seri

- b. *Analysis Filter Bank*

*Analysis Filter Bank* memiliki dua proses dimana dilakukan transformasi *Fourier* dan proses *filter* atau *bank filter*. Blok proses analisis *bank filter* ditunjukkan pada Gambar 2.14.



Gambar 2.14 Proses *Analysis Filter Bank* [10]

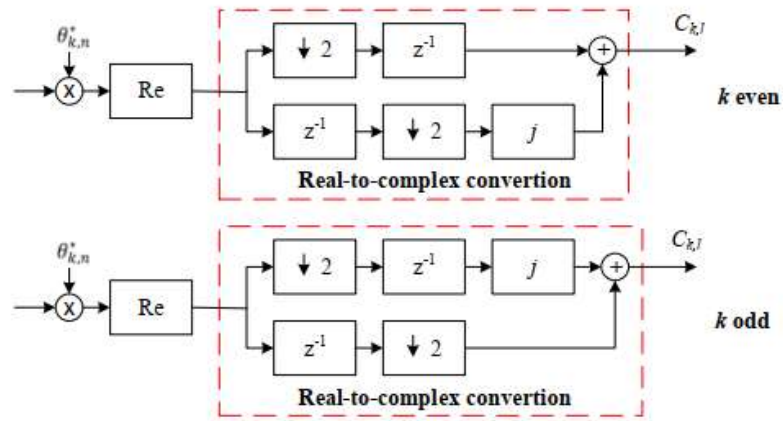


Gambar 2.15 Konfigurasi pada *Analysis Filter Bank* [10]

Berdasarkan Gambar 2.15, *analysis filter bank* melibatkan dua proses yakni proses FFT dan proses *filter*. *Filter* dalam *analysis filter bank* berfungsi untuk menjadikan satu sinyal sesuai dengan frekuensinya. Penggunaan jenis *filter* dalam *analysis filter bank* harus sesuai dengan jenis *filter* yang digunakan pada proses *syntesis filter bank* [10].

c. Pasca Pengolahan OQAM

Pasca pengolahan OQAM merupakan kebalikan dari pra pengolahan OQAM. Pada proses pengolahan OQAM terdiri dari dua operasi yaitu proses perkalian dengan kebalikan  $\theta_{kn}$  dan proses pengubahan bilangan riil menjadi bilangan kompleks. Pasca Pengolahan OQAM memiliki dua struktur yang sedikit berbeda tergantung pada nomor *sub*-kanal, seperti yang ditunjukkan pada Gambar 2.16



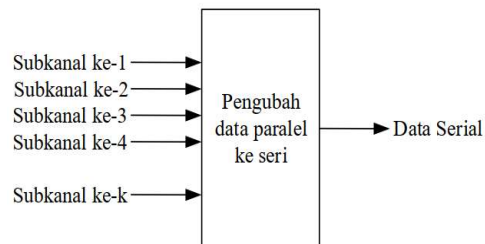
**Gambar 2.16 Pasca Pengolahan OQAM [10]**

d. Simbol *Demapping*

Pada sisi penerima, proses pemetaan ulang simbol merupakan proses kebalikan (*invers*). Pada proses ini, setiap simbol akan di *mapping* ulang agar terbentuk ukuran *bit* data yang sesuai perubahan tingkat modulasi. Dalam proses *mapping* ulang saat modulasi OQAM berfungsi untuk memilih simbol mana yang sebenarnya dikirim dari pemancar. *Mapping* dibutuhkan efek kanal dan derau (*noise*) yang mengakibatkan simbol OQAM dibagian penerima menjadi tidak sesuai aslinya dibandingkan dengan *output* dari pemetaan OQAM di pengirim. *Output* dari proses ini dalam bentuk *riil* dan imajiner yang selanjutnya dikembalikan sebagai bit data.

e. Pengubah Paralel menjadi Serial (P/S)

Pengubah paralel menjadi serial berfungsi untuk mengubah dari bit paralel asal ke bentuk serial dengan mengelompokkan *bit-bit* yang berasal dari keluaran pemetaan ulang menjadi satu baris sebagai keluaran dari proses pemetaan. Berikut blok pengubah paralel menjadi serial diperlihatkan pada Gambar 2.17.



### Gambar 2.17 Pengubah data paralel menjadi serial

Pada persamaan (2.8) model sistem komunikasi OQAM atau sistem FBMC. Pada pemancar simbol *input* kompleks ditulis sebagai[10]:

$$x_k(m) = a_k(m) + jb_k(m) \quad (2.8)$$

Dimana adalah  $a_k(m)$  dan  $b_k(m)$  adalah bagian simbol  $m$ th nyata dan imajiner pada *subcarrier*  $k$ , masing-masing, komponen dalam *fase* dan kuadratur mengatur bergiliran dalam domain waktu oleh  $T/2$ , dimana  $T$  adalah periode simbol. Simbol tersebut kemudian melewati transmisi *bank filter* dan dimodulasi menggunakan modulator *subcarrier*  $k$  yang frekuensi *carrier* nya adalah bagian  $1/T$ . Sinyal termodulasi OQAM-FBMC adalah [19]:

$$s(t) = \sum_{k=0}^{k-1} \sum_{m=-\infty}^{\infty} [a_k(m)h(t-mT) + jb_k(m)h(t-mT-T/2)]e^{jk\varphi_t} \quad (2.9)$$

Dimana  $h(t)$  adalah respons impuls dari prototipe *filter* dan  $\varphi_t = \frac{2\pi t}{T} + \frac{\pi}{2}$ , Setelah itu. Sinyal termodulasi OQAM-FBMC  $s(t)$  dimodulasi ke radio frekuensi (RF) dan ditransmisikan.

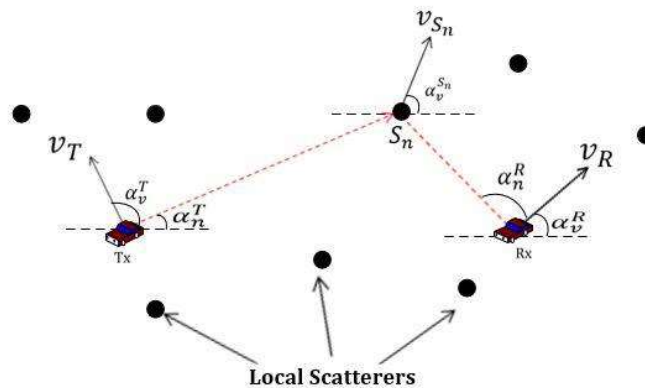
Untuk sistem transmisi yang ideal, sinyal yang diterima di penerima sama sinyal yang ditransmisikan di pemancar. Setelah demodulasi dari RF *Band*, Sinyal yang diterima  $r(t)$  didemodulasi menggunakan demodulator *subcarrier*  $K$  dan diteruskan ke *bank filter* yang cocok. Sinyal yang di *filter* kemudian diambil sampelnya dengan periode  $T$ , dan simbol keluarannya adalah [19]:

$$\hat{x}_k(m) = \hat{a}_k(m) + j\hat{b}_k(m) \quad (2.10)$$

dimana  $\hat{a}_k(m)$  dan  $\hat{b}_k(m)$  adalah bagian simbol  $m$ th nyata dan imajiner dari yang diterima pada *subcarrier*  $k$ .

## 2.2.6 Pemodelan Kanal dengan menggunakan V2V

Kanal V2V adalah model saluran yang didefinisikan secara geometris. *Transmitter* (Tx) dan *Receiver* (Rx) mengalami pergerakan dengan kecepatan tetap serta dikelilingi oleh *scatterers*. Dengan diasumsikan bahwa *transmitter* dan *receiver* bergerak beserta *scatterers* bergerak yang mempunyai kecepatan acak. Pada kanal V2V arah propagasi dibedakan menjadi dua bagian yakni sinyal ditransmisikan tanpa adanya penghalang LOS dan sinyal yang terkena *scatter* bergerak. Jumlah *scatterers* pada sisi pengirim dan penerima adalah berbeda. Jika sinyal dari sisi Tx terkena *scatter*, maka *scatterers* akan memantulkan sinyal tersebut ke sisi penerima [7].



Gambar 2.18 Skenario Propagasi Kanal V2V [7].

Pada Gambar 2.18 menunjukkan bahwa sistem komunikasi V2V *transmitter* (pemancar) dan *receiver* (penerima) dikelilingi oleh *scatterers* lokal yang dilambangkan dengan  $N$ , *scatter* lokal dilambangkan dengan  $S_n$  ( $n = 1, 2, \dots, N$ ), diasumsikan bahwa pemancar dan penerima bergerak dengan kecepatan konstan  $V_T$  dan  $V_R$  di arah yang ditentukan oleh sudut tetapnya  $\alpha_v^T$  dan  $\alpha_v^R$ . Masing-masing *scatterers* lokal bergerak dengan kecepatan  $v_{S_n}$  acak ke arah yang ditentukan oleh  $\alpha_v^{S_n}$ . Sinyal yang dipancarkan dari pemancar merambah ke arah *scatterers* kemudian dipantulkan ke semua arah demikian juga termasuk penerima. Sinyal masuk dari pemancar dilambangkan dengan sudut keberangkatan (AoD)  $\alpha_n^T$  mencapai ke



penerima dengan sudut kedatangan (AoA)  $\alpha_n^R$  setelah satu pantulan oleh  $S_n$  yang bergerak ke-n di dalam area propagasi seperti yang ditunjukkan pada Gambar 2.18 Dalam model ini penguatan saluran dihitung dengan mengikuti simbol [7]:

$$\mu(t) = \sum_{n=1}^N c_n e^{j(2\pi f_n t + \theta_n)} \quad (2.11)$$

Kanalnya disimbolkan dengan  $\mu(t)$  nilainya adalah  $n=1=N$  sejumlah  $n$ , jumlah *scatterers* kemudian dari faktor redaman yang dirumuskan sebagai  $c_n = \sigma_0 \sqrt{2/N}$  yang merupakan campuran dari semua redaman disebabkan oleh semua *scatterers*. Parameter  $\theta_n$  disebabkan pergeseran fasa dengan adanya *scatterers* di sekitar pemancar dan penerima. Sedangkan parameter  $f_n$  adalah total *Doppler Shift* yang disebabkan oleh *Doppler Shift* di sisi pemancar, penerima dan *scatterers*. Perhitungan parameter *Doppler Shift* ini dinyatakan sebagai berikut [7]:

$$f_n = f_n^T + f_n^{TS} + f_n^{SR} + f_n^R \quad (2.12)$$

Parameter  $f_n^T$  didefinisikan sebagai *Doppler Shift* dari pemancar. Parameter  $f_n^{TS}$  adalah *Doppler Shift* hasil dari interaksi antara pemancar dan *scatterers* yang bergerak.  $f_n^{SR}$  adalah *Doppler Shift* sebagai hasil dari interaksi antara *scatterers* dan penerima sedangkan  $f_n^R$  adalah *Doppler Shift* dari sisi penerima. Masing-masing Efek *Doppler* dirumuskan sebagai berikut [7]:

$$f_n^T = f_0 \frac{v_T}{c_0} \cos(\alpha_v^T - \alpha_n^T) \quad (2.13)$$

$$f_n^{TS} = f_0 \frac{v_T}{c_0} \cos(\alpha_v^T - \alpha_n^T) \quad (2.14)$$

$$\approx -f_0 \frac{v_{S_n}}{c_0} \cos(\alpha_n^T - \alpha_v^{S_n})$$

$$f_n^{SR} = (f_0 + f_n^{TS}) \frac{v_{S_n}}{c_0} \cos(\pi + \alpha_v^{S_n} - \alpha_n^R) \quad (2.15)$$

$$\approx f_0 \frac{v_{S_n}}{c_0} \cos(\alpha_v^{S_n} - \alpha_n^R)$$

$$f_n^R = (f_0 + f_n^{SR}) \frac{v_R}{c_0} \cos(\alpha_v^R - \alpha_n^R) \quad (2.16)$$

$$\approx f_0 \frac{v_R}{c_0} \cos(\alpha_v^R - \alpha_n^R)$$

Kemudian kesatuan akhir dari rumus sebelumnya menghasilkan bentuk final total *Doppler Shift*[7]:

$$f_n = \frac{k_0}{2\pi} \left[ \begin{array}{l} v_T \cos(\alpha_v^T - \alpha_n^T) - v_{S_n} + \\ (\cos(\alpha_n^T - \alpha_n^{S_n}) + \cos(\alpha_v^{S_n} - \alpha_n^R)) \\ + v_T \cos(\alpha_v^R - \alpha_n^R) \end{array} \right] \quad (2.17)$$

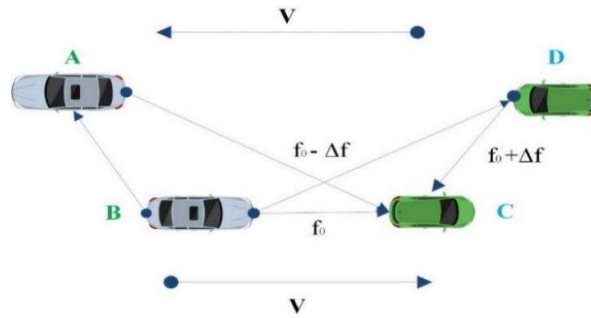
dengan parameter  $k_0 = 2\pi \frac{f_0}{c_0}$  adalah jumlah gelombang ruang bebas.

## 2.2.7 Efek *Doppler*

### a. *Doppler Shift*

Saat *transmitter* dan *receiver* dalam kondisi bergerak antara satu dan lainnya saling berkomunikasi, frekuensi *transmitter* mengalami perubahan karena adanya pergerakan dikedua sisinya. Jika *transmitter* dan *receiver* menghindari satu sama lain, frekuensi *transmitter* akan berkurang. Sebaliknya, jika *transmitter* dan *receiver* bergerak saling mendekati, maka perubahan frekuensi menjadi lebih besar. Fenomena ini disebut sebagai efek *Doppler*. Terjadinya efek *Doppler* digambarkan pada Gambar 2.11 yang menunjukkan bahwa, jika kendaraan B dan kendaraan C bergerak searah ke kiri dengan kecepatan (V) yang sama maka tidak terjadi efek *Doppler* ( $f_0$ ) atau tidak ada frekuensi *Doppler*[4]. Pada kendaraan A dan penerima C yang bergerak saling berlawanan dan menjauhi dengan V yang berbeda maka

menghasilkan *Doppler Shift* dengan perpindahan frekuensi  $f_0 - \Delta f$ . Sedangkan pada kendaraan C dan kendaraan D yang bergerak saling mendekati maka perpindahan frekuensi menjadi  $f_0 + \Delta f$ .



**Gambar 2.19 Doppler Shift [4]**

*Doppler shift* berarti frekuensi bergeser ke satu sisi disebabkan Gerakan receiver atau efek *Doppler*. *Doppler Shift* meningkatkan lebar pita sinyal yang dikirimkan. Sehingga dapat ditulis dengan menggunakan persamaan seperti dibawah ini[7]:

$$\Delta f = f_d = \pm f_c \frac{v}{c} \cos \beta \quad (2.18)$$

Dimana  $f_d$  adalah *Doppler Shift* dalam satuan (GHz),  $f_c$  adalah Frekuensi pembawa (5,8 GHz),  $v$  adalah kecepatan kendaraan (m/s),  $c$  adalah kecepatan gelombang elektromagnetik  $3 \times 10^8$  m/s dan  $\beta$  adalah sudut gerakan antara *transmitter* dan *receiver*. Persamaan tersebut menjelaskan bahwa untuk nilai  $\Delta f$  mempunyai nilai maksimum pada saat kondisi bernilai  $\beta = 0$ .

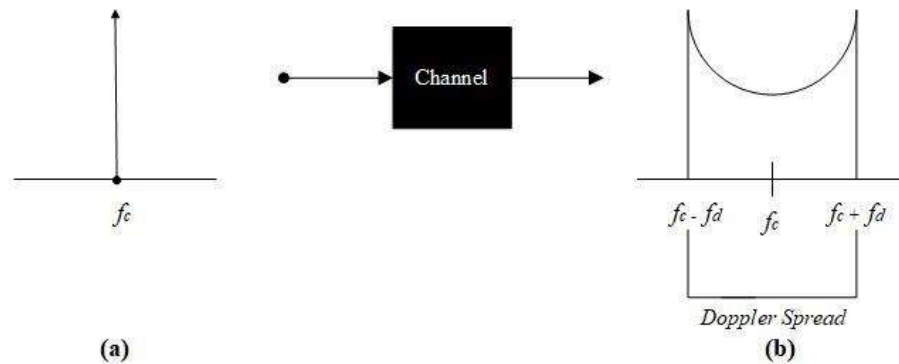
#### **b. Doppler Spread**

Pada Gambar 2.19 menunjukkan kondisi sebuah sistem komunikasi yang mempunyai sistem kinerja dalam rentang frekuensi  $f_c$  kemudian bergerak masuk ke dalam kanal dengan menggunakan pemodelan kanal *Vehicle to Vehicle*. Dari pergerakan tersebut maka terjadi pelebaran spektrum frekuensi (*Doppler spektrum*) karena adanya pengaruh efek *Doppler* dan diperparah dengan adanya *multipath*.

Pengaruh efek *Doppler* yang muncul pada pemodelan kanal seperti terlihat pada bagian (b). Apabila frekuensi  $f_c$  ditambahkan dengan  $f_d$  ( $f_c + f_d$ ) maka *spektrum* melebar ke bagian kanan. Sedangkan jika  $f_c$  dikurangi dengan  $f_d$  ( $f_c - f_d$ ) maka hasilnya *spectrum* melebar ke bagian kiri. Fenomena tersebut dinamakan *Doppler spread*. *Doppler spread* disebut juga sebagai pergeseran *Doppler shift* maksimum, karena kondisi *Doppler Shift* dapat meningkatkannya lebar pita sinyal yang dikirimkan [4]:

$$B_d = 2f_d \quad (2.19)$$

dengan  $B_d$  adalah *Doppler Spread* atau Lebar Pita Koheran (Hz) dan  $f_d$  merupakan frekuensi *Doppler* atau *Doppler Shift* (Hz).



**Gambar 2.20 Doppler Spread [4]**

### 2.2.8 Ekualisasi Linier

Ekualisasi MLSE memberikan kemampuan paling optimal diantara kategori ekualiser yang lain. Untuk respon kanal yang panjang, ekualiser MLSE menjadi sangat rumit oleh karena itu, dari sudut pandang implementasi praktis *filter* yang *sub* optimal menjadi pilihan. Ekualiser paling *sub* optimal adalah ekualiser linear yang memberikan kompleksitas lebih rendah. Ekualisasi linier sangat efektif untuk saluran dimana interferensi inter simbol tidak terlalu parah. Biasanya ekualiser di implementasikan sebagai struktur *filter* digital yang memiliki sejumlah tap dengan koefisien tap yang kompleks. Ekualiser dapat diklasifikasikan dengan berbagai cara antara lain [20]:

a. Pilihan berdasarkan adaptasi bobot tap ekualiser.

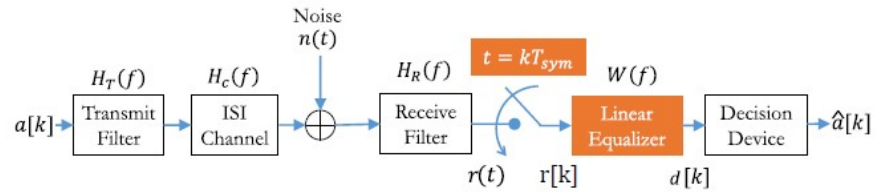
Berdasarkan tap koefisien adaptasi, ekualiser linier dibagi menjadi 2 jenis yaitu:

1. *Non-Adaptive Equalizer*: Jika sebuah kanal menggunakan kanal yang *time invariant* selama *interval* transmisi data, bobot ekualiser diperhitungkan pada saat sesi transfer yang dimana ditentukan selama sisa saat sesi transfer.
2. *Adaptive Equalizer*: Untuk kanal *time varian*, koefisien pada *adaptive equalizer* dapat berubah terhadap waktu dikalkulasi dengan algoritma yang dapat menyesuaikan dengan variasi kondisi kanal. Terdapat opsi untuk mengkalkulasi serta membuat baru bobot adaptif, termasuk kuadrat rata-rata terkecil, kuadrat kisi rekursif, *Kalman* konvensional, *Kalman* akar kuadrat, dan *Fast Kalman*.

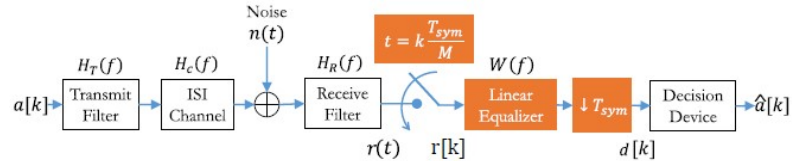
b. Pilihan berdasarkan durasi proses mengambil *sampling*.

Proses memilih durasi dalam *sampling* dengan durasi simbol mempengaruhi desain dan kinerja ekualizer. Jadi terdiri dari dua jenis Teknik ekualisasi antar lain:

1. Ekualiser linier dengan spasi simbol: Hal ini diilustrasikan pada Gambar 2.21, yang mana *downsampler* di depan ekualiser, beroperasi pada satu contoh per waktu simbol ( $T_{sym}$ ). Ekualiser ini memiliki tingkat sensitifitas terhadap waktu *sampling*nya yang dapat rusak akibat *aliasing*, Sehingga kinerja mereka tidak akan bisa optimal.
2. Ekualiser linier dengan jarak fraksional: Bertujuan agar jarak antara ekualizer berkurang dan berdekatan menjadi Sebagian kecil dari interval simbol. Demikian pula, sebelum ekualizer, *subsampling* dilakukuan pada laju yang lebih besar daripada laju simbol (untuk beberapa bilangan bulat  $k$  dan  $M$ , periode  $kT_{sym}=M$ ). Keluaran ekualiser dikurangi dengan simbol waktu sebelum mengirim simbol keluaran detector. Hal ini ditunjukkan oleh Gambar 2.22. Walaupun lebih banyak perhitungan yang diperlukan, ekualiser fraksional menyederhanakan desain demodulator lainnya serta dapat menggantikan waktu *sample* yang mengalami perubahan.



Gambar 2.21 Ekualiser Linier Spasi Simbol pada saluran waktu kontinu



**Gambar 2.22 Ekualiser Linier Berspasi Fraksional pada saluran waktu kontinu**

- c. Pilihan berdasarkan kriteria untuk pengoptimalan pengaruh tap.

Bobot tap ekualiser biasanya dipilih berdasarkan beberapa kriteria optimal yang dua di antaranya adalah:

1. Kriteria distorsi puncak: Memiliki tujuan meminimalkan distorsi semaksimal mungkin dikarenakan adanya ISI, pada keluaran ekualiser. Ini juga mendekati dengan kriteria *Zero Forcing*. Ini merupakan dasar dari untuk ekualiser *Zero Forcing*.
2. Kriteria *Mean Square Error* (MSE) : Bertujuan agar menjadikan minim *root mean square* dari *error term* yang dikalkulasi dari perbedaan antar keluaran ekualiser dan simbol informasi yang akan dikirim. Ini adalah dasar dari ekualiser *Linear Minimum Mean Square Error* (LMMSE) dan algoritma *Least Mean Square* (LMS)

### 2.2.8.1 Ekualisasi *Zero Forcing*

Sebuah ekualiser *Zero Forcing* (ZF) dinamakan demikian karena memaksa sisa ISI dalam *output* ekualiser  $d[k]$  menjadi nol. Ekualiser pemaksaan nol yang optimal mencapai pemerataan sempurna dengan memaksa sisa ISI pada saat pengambilan sampel  $kT$  kecuali  $k=0$ . Tujuan ini dapat dicapai jika kita mengizinkan penyeimbang dengan struktur *Infinite Impulse Response* (IIR). Dalam sebagian besar aplikasi praktis, fungsi transfer saluran  $H(z)$  dapat didekati dengan

filter *Finite Response* (FIR) dan oleh karena itu, pasangan penyamaannya yang sempurna  $W(z)$  akan menjadi *filter* IIR [20].

$$W(z) = \frac{1}{H(z)} \quad (2.20)$$

akibatnya, fungsi transfer keseluruhan yang dihasilkan, dilambangkan dengan  $Q(z)$ , diberikan oleh:

$$Q(z) = H(z)W(z) = 1 \quad (2.21)$$

sebaliknya, dalam domain waktu, solusi *Zero-Forcing* memaksa respons keseluruhan menjadi nol di semua posisi kecuali untuk posisi di  $k_0$  dimana nilainya sama dengan fungsi *delta Dirac*.

$$q[k] = h[k] * w[k] = \delta[k - k_0] = \begin{cases} 1, & \text{untuk } k = k_0 \\ 0, & \text{untuk } k \neq k_0 \end{cases} \quad (2.22)$$

Untuk implementasi praktis, karena stabilitas yang melekat dan ketahanan yang lebih baik terhadap efek *length word* yang terbatas, *Filter* FIR lebih disukai daripada *filter* IIR. Menerapkan ekualiser *Zero Forcing* sebagai *filter* FIR berarti memaksakan kausalitas dan batasan panjang pada *filter* ekualiser yang fungsi transfernya berbentuk [20]:

$$W(z) = \sum_{k=0}^{N-1} w[k] z^{-k} \quad (2.23)$$

Mengingat kanal respons impuls dengan panjang  $L$  dan filter ekualiser dengan panjang  $N$ , solusi *Zero Forcing* dalam persamaan 2.22, dapat dinyatakan sebagai berikut:

$$q[k] = h[k] * w[k] = \sum_{i=-\infty}^{\infty} h[i] w[k-i] = \delta[k - k_0], \quad k = 0, L, L + N - 2 \quad (2.24)$$

secara sederhana, dengan solusi *zero-forcing*, respons impuls keseluruhan mengambil bentuk berikut:

$$q[k] = h[k] * w[k] = \delta[k - k_0] = [0, 0, L, 0, 1, 0, L, 0, 0] \quad (2.25)$$

di mana,  $q[k] = 1$  pada posisi  $k_0$ . Jumlah konvolusi dari barisan panjang berhingga  $h$  dari panjang  $L$  dan urutan panjang berhingga kausal  $w$  dengan panjang  $N$  dapat dinyatakan dalam bentuk matriks sebagai (2.26) [20]:

$$\begin{pmatrix} q[0] \\ q[1] \\ q[2] \\ \vdots \\ M \\ q[L+N-2] \end{pmatrix} = \begin{pmatrix} h[0] & h[-1] & L & h[-(N-1)] \\ h[1] & h[0] & L & h[-(N-2)] \\ h[2] & h[1] & L & h[-(N-3)] \\ \vdots & \vdots & \vdots & \vdots \\ M & M & O & M \\ h[L+N-2] & h[L+N-3] & L & h[L-1] \end{pmatrix} \cdot \begin{pmatrix} w[0] \\ w[1] \\ w[2] \\ \vdots \\ M \\ w[N-1] \end{pmatrix} \quad (2.26)$$

maka, solusi *Zero Forcing* dalam persamaan 2.36 dapat dinyatakan dalam bentuk matriks sebagai:

$$\begin{pmatrix} h[0] & h[-1] & L & h[-(N-1)] \\ h[1] & h[0] & L & h[-(N-2)] \\ h[2] & h[1] & L & h[-(N-3)] \\ \vdots & \vdots & \vdots & \vdots \\ M & M & O & M \\ h[L+N-2] & h[L+N-3] & L & h[L-1] \end{pmatrix} \cdot \begin{pmatrix} w[0] \\ w[1] \\ w[2] \\ \vdots \\ M \\ w[N-1] \end{pmatrix} = \begin{pmatrix} 0 \\ M \\ 0 \\ 1 \\ 0 \\ M \\ 0 \end{pmatrix} \quad (2.27)$$

$$H.w = \delta_{k_0}$$

Jika diasumsikan kausalitas dari respon impuls saluran,  $h[k] = 0$  untuk  $k < 0$  dan jika  $L$  dipilih cukup besar  $h[k] \approx 0$  berlaku untuk  $k \geq L$ . Maka, solusi *zero-forcing* dapat disederhanakan sebagai (2.28) [20]:

$$\begin{pmatrix} h[0] & 0 & L & 0 & 0 \\ h[1] & h[0] & L & 0 & 0 \\ h[2] & h[1] & L & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M & M & O & M & M \\ 0 & 0 & O & h[L-1] & 0 \\ 0 & 0 & L & 0 & h[L-1] \end{pmatrix} \cdot \begin{pmatrix} w[0] \\ w[1] \\ w[2] \\ \vdots \\ M \\ w[N-2] \\ w[N-1] \end{pmatrix} = \begin{pmatrix} 0 \\ M \\ 0 \\ 1 \\ 0 \\ M \\ 0 \end{pmatrix} \quad (2.28)$$

yang dapat dinyatakan dalam matriks *Toeplitz*,  $H = T(h)$

$$\begin{aligned} T(h).w &= \delta_{k_0} \\ H.w &= \delta_{k_0} \end{aligned} \quad (2.29)$$

Jika ekualiser *Zero Forcing* diasumsikan secara sempurna mengkompensasi respon impuls saluran, *output* dari ekualiser akan menjadi fungsi delta *Dirac* yang tertunda oleh simbol tertentu karena penundaan ekualiser ( $k_0$ ). Dalam bentuk matriks, kondisi ini direpresentasikan sebagai (2.30) [20]:



$$H.w = \delta_{k_0} \quad (2.30)$$

Dengan simbol H merupakan matriks dari saluran persegi panjang yang dijabarkan dalam persamaan 2.28 dan 2.30, w merupakan vektor kolom yang berisi bobot tap ekualiser dan  $\delta_{k_0}$  adalah vektor kolom yang mewakili fungsi delta *Dirac*, dengan semua elemen sama dengan nol kecuali untuk elemen kesatuan di posisi ( $k_0$ ). Posisi elemen kesatuan dalam matriks  $\delta_{k_0}$  menentukan *delay* ekualiser ( $k_0$ ). Penundaan ekualiser memungkinkan adanya *delay* pada sistem ujung ke ujung. Tanpa penundaan ekualiser, akan sulit untuk mengkompensasi penundaan saluran efektif menggunakan ekualiser FIR kausal [20].

### 2.2.9 Bit Error Rate (BER)

*Bit Error Rate* (BER) adalah suatu rumusan kesalahan laju bit (*bit error*) yang terjadi selama pentransmisi data antara terminal pengirim dan terminal penerima. Kesalahan yang terjadi selama proses pentransmisi data antara data masukan dan keluaran dengan cara membandingkan antara data masukan total yang dikirim dengan jumlah data keluaran yang rusak. Setelah proses demodulasi dengan cara membandingkan hasil keluaran dengan hasil masukan. *Bit Error Rate* tersebut dapat dihitung dengan persamaan (2.31) [21]:

$$BER = \frac{\text{jumlah\_bit\_error}}{\text{jumlah\_total\_bit\_kirim}} \quad (2.31)$$

Adapun cara untuk menghitung probabilitas rasio kesalahan bit untuk modulasi M-QAM menggunakan kanal AWGN dapat dihitung menggunakan persamaan (2.32) dibawah ini [22]:

$$P_B = \frac{4}{n} \left( 1 - \frac{1}{\sqrt{M}} \right) Q \left( \sqrt{\frac{3n}{M-1}} \cdot \frac{E_b}{N_o} \right) \quad (2.32)$$

Atau bisa menggunakan persamaan (2.33) [22]:

$$P_B = \frac{2}{ld(M)} \left( 1 - \frac{1}{\sqrt{M}} \right) \text{erfc} \left( \sqrt{\frac{3ld(M)}{2(M-1)}} \cdot \frac{E_b}{N_o} \right) \quad (2.33)$$

Dimana:

$E_b$  = energi per bit

$N_o$  = *noise power spectral density*

$N$  =  $\log_2(M) = \log_2(M)$  = jumlah bit per simbol

$M$  =  $2^n$  = jumlah simbol yang berbeda untuk modulasi atau ukuran konstelasi modulasi