

BAB III

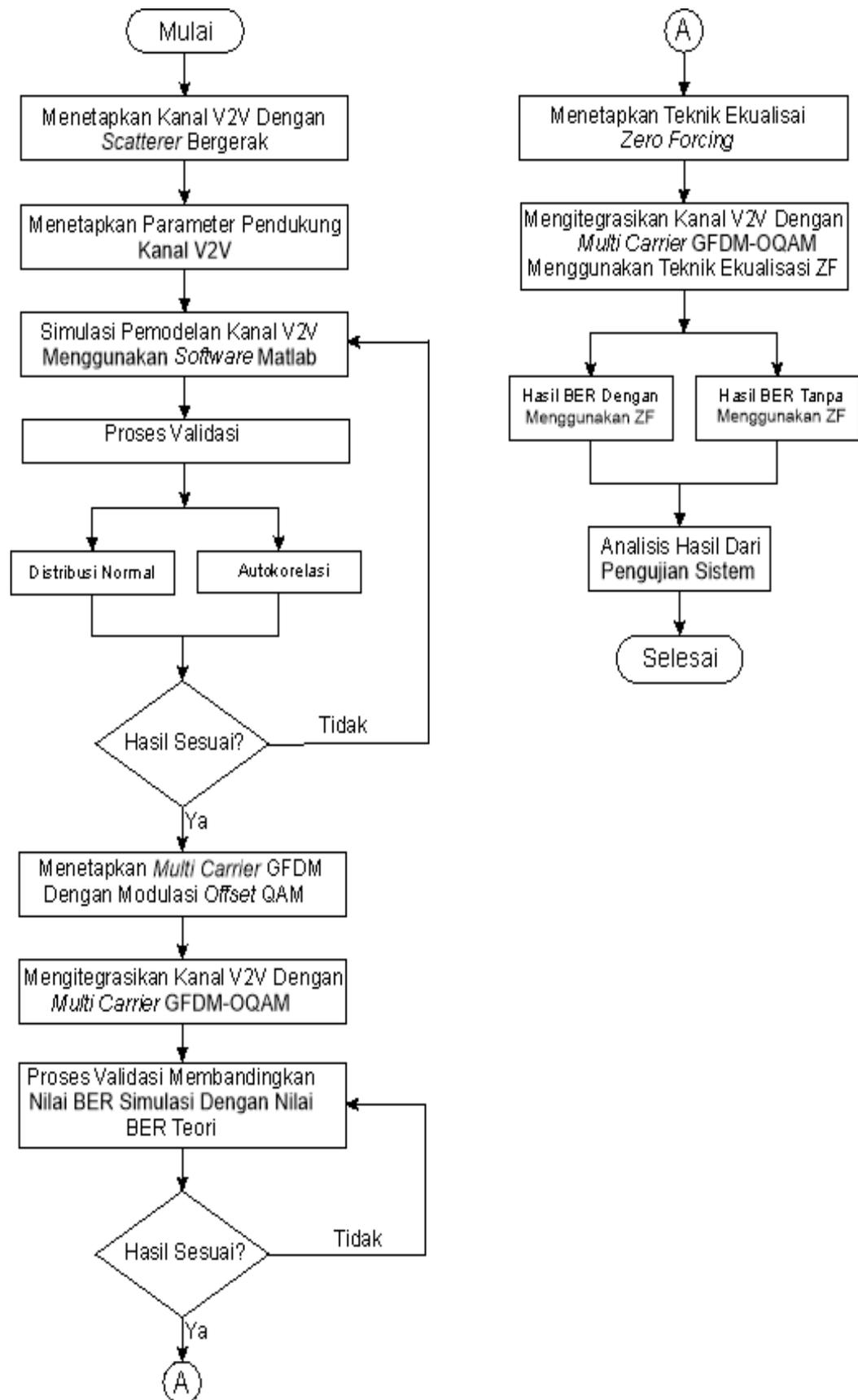
METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

Pada penelitian ini menggunakan kanal *vehicle to vehicle* dengan *scatterer* bergerak. Kecepatan yang digunakan sebesar 17 m/s, 22 m/s dan 30 m/s pada sisi *transmitter* dan *receiver*, sedangkan pada sisi *scatterer* sebesar 5 m/s. Jumlah *scatterer* yang digunakan adalah 8. Sistem *multi carrier* yang digunakan adalah GFDM. Untuk memitigasi efek *Doppler* yang terjadi dengan menambahkan metode ekualisasi *Zero Forcing* dan memvalidasi hasil integrasinya dalam parameter *Bit Error Rate*. Kemudian untuk mensimulasikan parameter yang digunakan pada penelitian ini akan digunakan *software* Matlab R2018a.

3.2 ALUR PENELITIAN

Penelitian ini dilakukan dalam beberapa tahap, yaitu tahap pertama penulis menetapkan model kanal, model kanal yang digunakan adalah kanal V2V dengan *scatterer* bergerak. Dengan menambahkan parameter pendukung seperti jumlah *scatterer* yang bergerak adalah 8, variasi kecepatan yang digunakan sebesar 17 m/s, 22 m/s dan 30 m/s pada sisi *transmitter* dan *receiver*, sedangkan pada sisi *scatterer* sebesar 5 m/s. Serta ditambahkan *noise* yaitu *noise* AWGN. Tahap kedua yaitu simulasi pemodelan kanal menggunakan *software* Matlab. Tahap ketiga yaitu proses validasi, proses validasi dilakukan dengan menggunakan distribusi normal dan autokorelasi. Apabila hasil validasi belum sesuai, maka kembali ke tahap simulasi pemodelan kanal V2V dengan *scatterer* bergerak dan tahap validasi hasil simulasi. Apabila hasil dari proses validasi berhasil sesuai dengan teori, maka hasil tersebut digunakan untuk tahap keempat yaitu menentukan sistem *multi carrier* yang akan digunakan yaitu *Generalized Frequency Division Multiplexing* (GFDM) yang dimodulasi oleh *Offset* QAM dengan menggunakan modulasi dasar 16-QAM. Tahap kelima adalah mengintegrasikan model kanal V2V dengan *scatterer* bergerak dengan sistem *multi carrier* GFDM. Tahap keenam yaitu proses validasi dari hasil integrasi kanal V2V dengan *multi carrier* GFDM. Pada Gambar 3.1 merupakan *flowchart* alur penelitian yang dilakukan.



Gambar 3.1 *Flowchart* Alur Penelitian

Proses validasi dilakukan dengan cara membandingkan nilai BER hasil dari simulasi menggunakan Matlab dengan nilai BER teori. Apabila hasil dari proses validasi tidak sesuai dengan teori, maka kembali ke tahap simulasi mengintegrasikan model kanal V2V dengan *scatterer* bergerak dengan sistem *multi carrier* GFDM dan tahap validasi hasil simulasi. Apabila hasil dari proses validasi berhasil sesuai dengan teori, maka hasil tersebut digunakan untuk tahap kedelapan yaitu menentukan teknik ekualisasi. Ekualisasi yang digunakan adalah ekualisasi *Zero Forcing*. Hasil integrasi kanal V2V dengan *multi carrier* GFDM dilakukan teknik ekualisasi untuk memitigasi efek *Doppler* yang terjadi. Setelah ekualisasi *Zero Forcing* diterapkan. Tahap selanjutnya adalah membandingkan nilai BER yang diberi ekualisasi *Zero Forcing* dengan nilai BER yang tanpa diberi ekualisasi *Zero Forcing*. Tahap terakhir adalah menganalisis pengaruh kecepatan *scatterer* yang bergerak atau efek *Doppler* terhadap parameter *Bit Error Rate* yang dihasilkan dengan kanal V2V menggunakan *multi carrier* GFDM dengan teknik ekualisasi *Zero Forcing*.

3.2.1 Penetapan Kanal

Proses pertama yang penulis lakukan adalah penetapan kanal transmisi. Hal ini untuk mendukung penelitian yang bertautan dengan simulasi pemodelan yang diaplikasikan pada suatu kanal untuk menganalisis parameter final yaitu *Bit Error Rate* (BER) dari hasil simulasi. Kanal yang digunakan pada penelitian ini yaitu *Vehicle to Vehicle* (V2V). Kanal *Vehicle to Vehicle* merupakan sebuah pemodelan kanal yang ditentukan dengan model geometri. Pengirim (Tx) dan Penerima (Rx) bergerak dengan kecepatan konstan dan dikelilingi oleh *scatterer*. Untuk memvalidasi kanal tersebut dilakukan dengan proses distribusi dan autokorelasi, distribusi yang digunakan adalah distribusi normal (*Gaussian*). Selanjutnya akan dilakukan simulasi dengan menggunakan *multicarrier* GFDM-OQAM dan ekualisasi *Zero Forcing*.

3.2.2 Parameter Simulasi

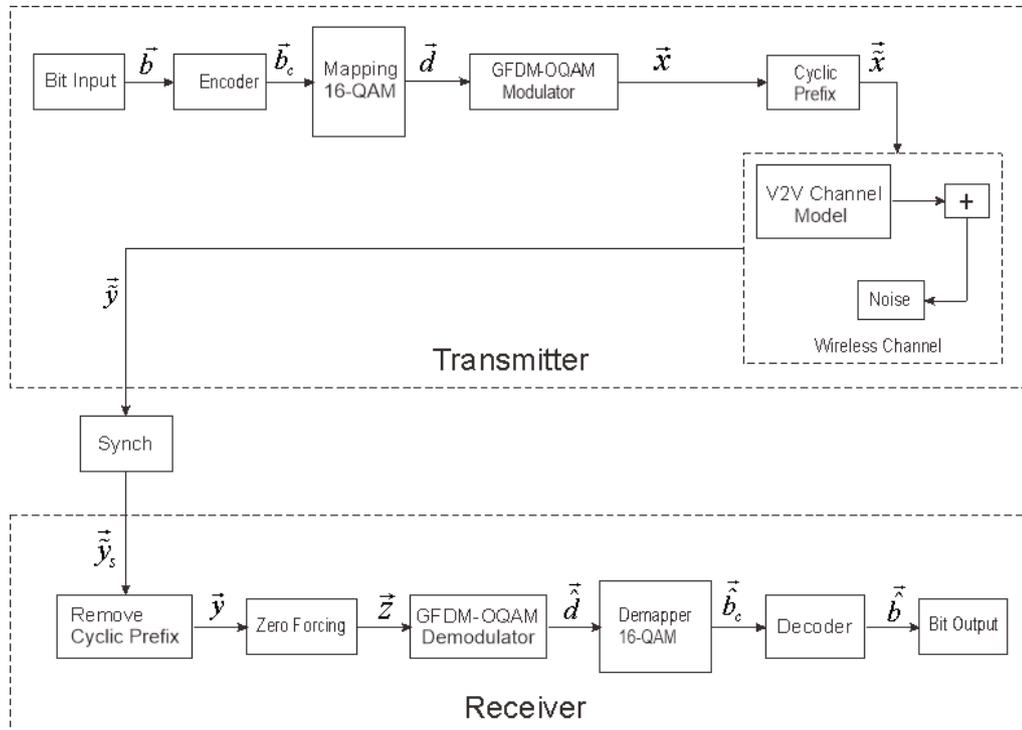
Pada simulasi ini peneliti menggunakan *multicarrier* GFDM dengan modulasi digital *Offset-QAM*, parameter modulasi digital yang digunakan adalah *Pulse shaping filter*, *Eb/No*, *Bit Error Rate (BER)*, dan kanal *Vehicle to Vehicle (V2V)*. Kecepatan yang digunakan pada kanal V2V baik pada sisi pengirim, penerima dan *scatterer* menggunakan kecepatan yang bervariasi. Untuk proses validasi menggunakan distribusi *Gaussian* (normal) dan fungsi autokorelasi. Jenis *Pulse Shaping Filter* yang digunakan adalah *Root Raised Cosine*. Pada modulasi digital menggunakan 16-QAM yang kemudian dilakukan proses *offset* sehingga menjadi *Offset-QAM (OQAM)*. Untuk menunjang hasil analisa, dilakukan perbandingan unjuk kerja pada parameter BER yang dihasilkan dari simulasi terhadap *Eb/No* yang nilainya dari 0 dB sampai dengan 15 dB dengan menggunakan ekualisasi *Zero Forcing (ZF)* dan tanpa menggunakan ekualisasi *Zero Forcing (ZF)* pada sisi penerima dengan kecepatan yang bervariasi. Untuk parameter simulasi dapat dilihat pada Tabel 3.1.

Tabel 3.1 Parameter Simulasi

Parameter	Simbol	Nilai
Jenis Modulasi	-	16-QAM
Jumlah Level Modulasi	ml	4
Jumlah <i>Scatterer</i>	N	8
Kecepatan <i>Scatterer</i>	vs	5 m/s
Kecepatan kendaraan pengirim	vt	17 m/s; 22 m/s dan 30 m/s
Kecepatan kendaraan penerima	vs	17 m/s; 22 m/s dan 30 m/s
Sudut antara garis horizontal dengan arah pergerakan kendaraan Tx	α_{vt}	5°
Sudut antara garis horizontal dengan arah pergerakan kendaraan Rx	α_{rv}	5°
Fase	θ	5
<i>Subcarrier</i>	K	5
<i>Subsymbol</i>	M	9
<i>Bit Input</i>	b	180000
Kanal	h	V2V
<i>Pulse Shaping Filter</i>	g	<i>Root raised cosine</i>
<i>Roll-off factor</i>	α	1
<i>Frequency Carrier</i>	Fc	5,8 GHz
<i>Frequency Sampling</i>	Fs	500 Hz
Jumlah simbol	j	1000

3.2.3 Pemodelan Sistem

Pemodelan sistem utama GFDM dengan skema modulasi *offset* QAM, peneliti merujuk pada [6], gambar pemodelan sistem dapat dilihat pada Gambar 3.2 berikut:



Gambar 3.2 Pemodelan Sistem Utama GFDM-OQAM

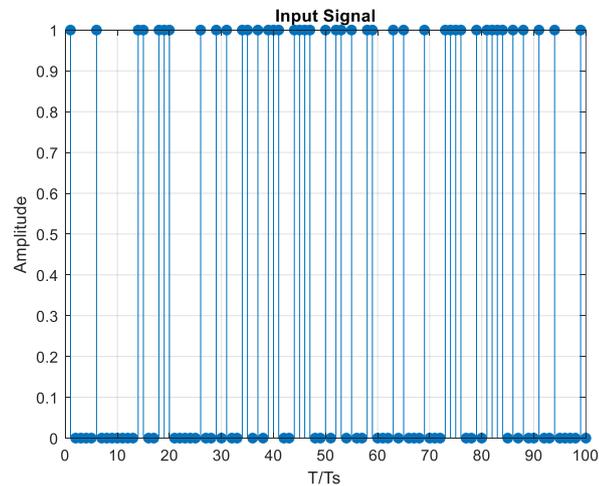
3.2.3.1 Bagian Transmitter

Pada bagian *transmitter* ini terdiri dari beberapa blok, antara blok satu dengan blok lainnya saling terhubung, blok yang saling terhubung antara lain adalah bit *input*, *Encoder*, *Mapping* 16-QAM, GFDM Modulator, *cyclic prefix*, dan *channel Vehicle to Vehicle*. Proses pertama data masukan berupa bit, kemudian dibangkitkan dengan sistem acak sejumlah 180000 bit. Selanjutnya bit yang telah dibangkitkan dikodekan terlebih dahulu, setelah itu bit yang sudah dikodekan, akan mengalami proses modulasi melalui *modulator*.

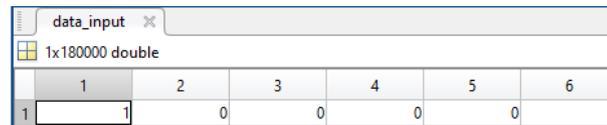
1) Bit Input

Proses pertama yang peneliti lakukan adalah *bit input* (\vec{b}) di mana pada penelitian ini menggunakan data masukan berupa bit yang dibangkitkan secara acak dengan nilai 0 dan 1, dengan *subcarrier* berjumlah 5 dan *subsymbol* berjumlah 9

dengan jumlah simbol senilai 1000, sehingga total bit informasi yang digunakan berjumlah 180000 bit. Pada Gambar 3.3 menampilkan 100 bit pertama sinyal informasi dari data *bit* yang dibangkitkan secara acak. Hasil dari pembangkitan sinyal disimbolkan dengan \vec{b} , untuk bit biner dapat dilihat pada Gambar 3.4.



Gambar 3.3 Sinyal *Input*



Gambar 3.4 Bit Biner Sinyal *Input*

2) *Encoder*

Setelah melalui proses pembangkitan bit pada data \vec{b} selanjutnya dilakukan proses pengkodean untuk mendapatkan data \vec{b}_c . Proses ini merupakan proses *encoder* di mana data \vec{b} dikonversi ke *M-ary* desimal level, dengan $M=2^n$. Proses ini membagi urutan bit menjadi *n-bit* sesuai dengan modulasi yang digunakan. Pada penelitian ini digunakan 180000 bit secara serial kemudian dikonversikan menjadi bentuk paralel sebanyak 45000 bagian, yang mana setiap bagian memiliki 4 bit yang berbeda karena menggunakan modulasi 16-QAM. Maka *output* yang dihasilkan \vec{b}_c menjadi 45000×4 bagian, seperti yang terlihat pada Gambar 3.5.

	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	0	0
4	0	1	1	0
5	0	1	1	1

Gambar 3.5 Hasil Bit *Encoder*

3) *Mapping 16-QAM*

Proses *mapping* berfungsi mengubah keluaran dari *encoder* yang mana setiap simbol terdiri dari bit biner menjadi bentuk simbol baru yang berbentuk bilangan kompleks. Setiap simbol acak yang sudah dibangkitkan selanjutnya dipetakan dengan proses modulasi 16-QAM. Terdapat dua bagian pada modulasi QAM yaitu *inphase* (I) atau bilangan riil, dan *quadrature* (Q) atau bilangan imajiner. Modulasi 16-QAM artinya setiap simbol yang diterima terdiri dari 4 bit dengan kemungkinan 16 kombinasi simbol. Untuk proses pengubahan biner menjadi bentuk simbol kompleks sebagai berikut:

Pada modulasi 16-QAM, simbol pertama adalah 1000, maka nilai $b(i) = 0, b(i+1)=0, b(i+2)=0, b(i+3)=1$.

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

$$x = \frac{1}{\sqrt{10}} \left\{ 1 - 2(1) [2 - (1 - 2(0))] + j(1 - 2(0)) [2 - (1 - 2(0))] \right\}$$

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

$$x = \frac{1}{\sqrt{10}} \{1 + j3\}$$

$$x = 0,3162 + 0,9487i$$

Simbol kedua adalah 0100, maka nilai $b(i) = 0, b(i+1)=0, b(i+2)=1, b(i+3)=0$

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

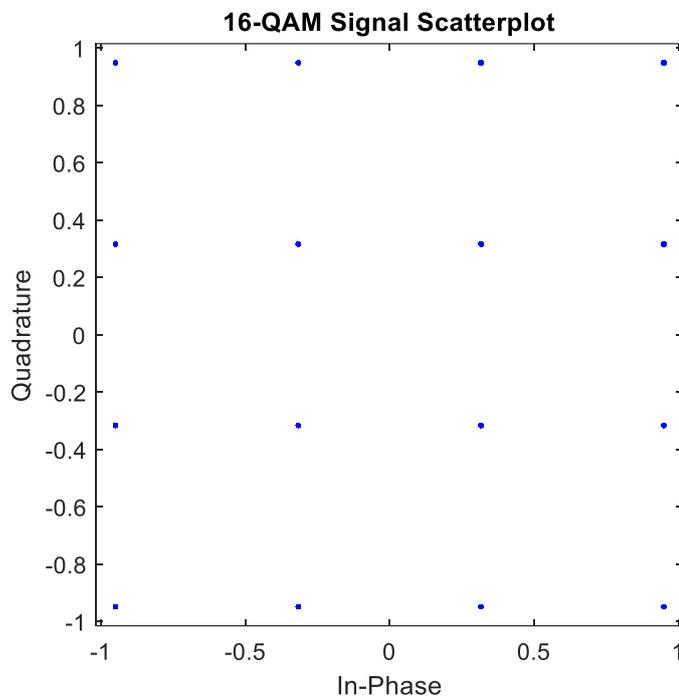
$$x = \frac{1}{\sqrt{10}} \left\{ 1 - 2(0) [2 - (1 - 2(1))] + j(1 - 2(0)) [2 - (1 - 2(0))] \right\}$$

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

$$x = \frac{1}{\sqrt{10}} \{3 + j1\}$$

$$x = 0,9487 + 0,3162i$$

Dilihat dari dasar perhitungan pemetaan 16-QAM yang dilakukan, jika hasil biner 1000 dan 0100 ditempatkan pada titik-titik diagram konstelasi akan menempati posisi seperti yang terlihat pada Gambar 3.6. sedangkan untuk perhitungan dengan bit simbol 1000 dan 0100 menghasilkan bilangan kompleks senilai $0,3162+0,9487i$ dan $0,9487+0,3162i$, seperti yang terlihat pada Gambar 3.7.



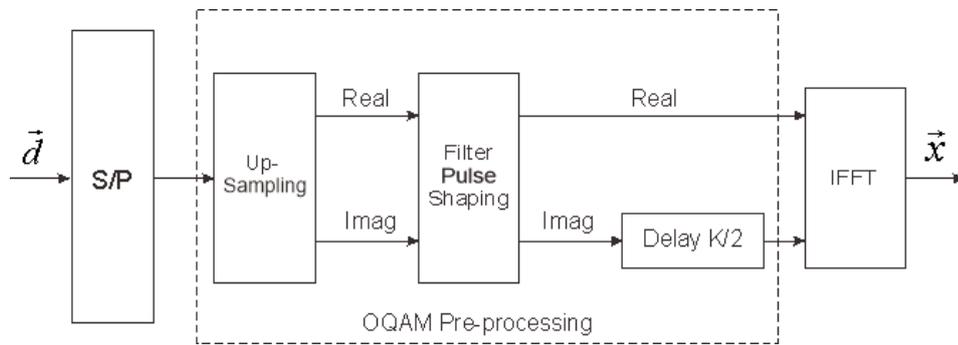
Gambar 3.6 Konstelasi 16-QAM

qammod2			
1x45000 complex double			
	1	2	3
1	0.3162 + 0.9487i	0.9487 + 0.3162i	0.3162 + 0.3162i

Gambar 3.7 Mapping Simbol 16-QAM

4) *Modulator GFDM-OQAM*

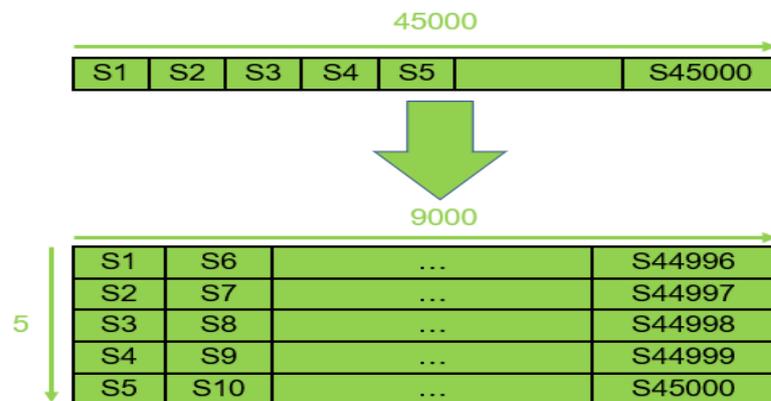
Berikut Gambar 3.8 merupakan blok diagram *modulator* GFDM-OQAM.



Gambar 3.8 Blok Diagram *Modulator* GFDM-OQAM

a) *Serial to Parallel*

Setelah melalui proses *mapping*, selanjutnya adalah proses *serial to parallel converter* bertujuan mengubah sederetan simbol berbentuk *serial* menjadi simbol berbentuk *parallel* yang telah dimodulasi sebelumnya. Simbol awal berbentuk serial dengan nilai 1×45000 , kemudian matriks \vec{d} serial ini dikonversi ke dalam bentuk *parallel* menjadi 5×9000 . Hal ini sesuai dengan jumlah *subcarrier* yang digunakan sebanyak 5 *subcarrier*, seperti yang diilustrasikan pada Gambar 3.9.



Gambar 3.9 Konversi *Serial to Parallel*

b) *Up-Sampling*

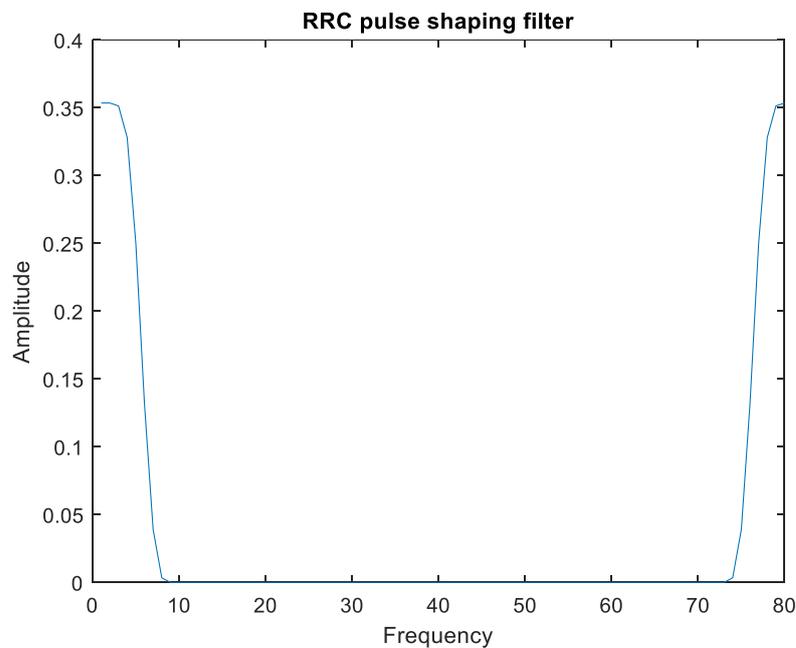
Hasil dari *mapping* 16-QAM (matriks \vec{d}) dengan skala 8×10 ($K \times M$) berisi deretan data informasi. Sisa dari matriks pada blok GFDM yang tidak terisi oleh data informasi akan bernilai nol (0), nilai nol tersebut digunakan untuk proses *circular convolution*. Hasil dari *Up-sampling* dengan simulasi dapat dilihat pada Gambar 3.10.

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0

Gambar 3.10 Hasil *Up-sampling*

c) **Filter Pulse Shaping**

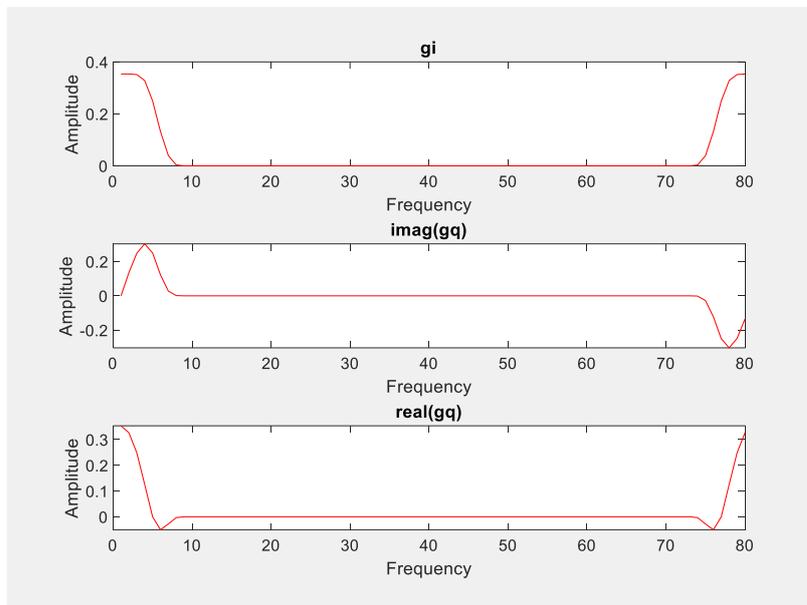
Proses selanjutnya adalah memproses filter *pulse shaping*. Jenis *pulse shaping* yang digunakan adalah *Root Raised Cosine* (RRC). *Pulse shape* yang dibentuk pada domain waktu akan dikalikan dengan hasil dari sistem GFDM. Pada penelitian ini besaran *roll-off factor* yang digunakan adalah 1. Hasil dari *pulse* RRC diberikan simbol g , selanjutnya untuk menghasilkan *pulse shaping*, g digeser secara sirkular pada domain waktu dan frekuensi. Filter *pulse shaping* menghasilkan frekuensi dan waktu yang berbeda, hal ini dikarenakan adanya pergeseran secara sirkular. Hasil dari proses *Up-sampling* dengan matriks 8×10 ($K \times M$) dirubah menjadi 80×1 yang mewakili satu buah simbol GFDM. Pada Gambar 3.11 adalah bentuk sinyal g dengan $\alpha = 1$ pada domain frekuensi.



Gambar 3.11 Sinyal QAM dengan *pulse shaping* RRC dengan $\alpha = 1$

d) *Offset QAM*

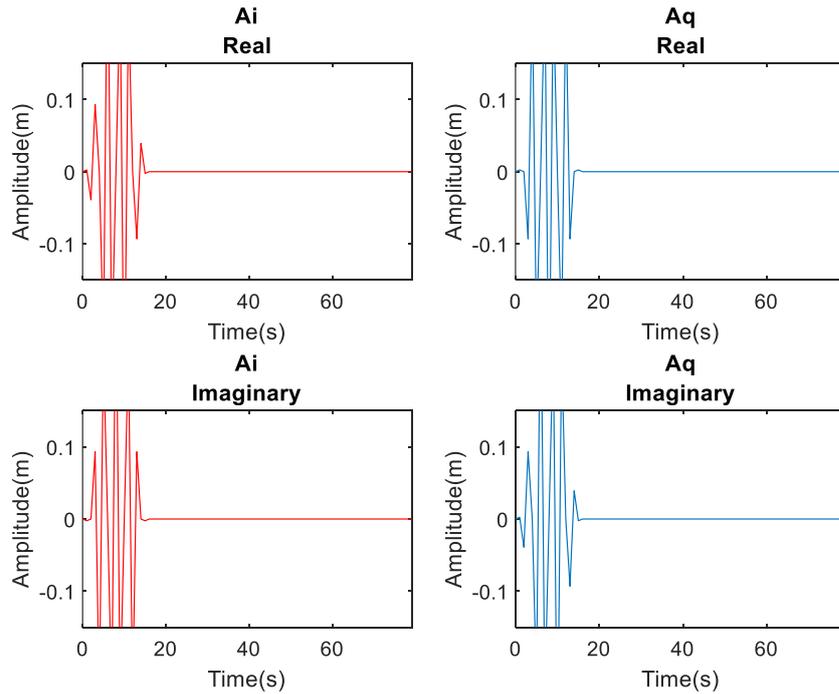
Selanjutnya adalah proses *offset QAM* dari filter *pulse shaping* $g[n]$ yang telah dibentuk sebelumnya, di sini terjadi proses dengan mentransfer bagian real dan imajiner dari data simbol QAM menggunakan *pulse shaping* dengan pergeseran $K/2$ cuplikan atau setengah *subcarrier* satu dengan yang lainnya pada sisi *quadrature*. Lalu rotasi *phasa* dilakukan sebesar $\pi/2$ antar *subsymbol* dan *subcarrier* yang berdekatan. Proses *Offset QAM* ini bertujuan untuk membuat transmisi *free interference* dapat tercapai. Hal ini yang membedakan antara QAM dan *Offset QAM* pada sistem GFDM. Pada Gambar 3.12 adalah sisi *inphase* dan *quadrature* pada *Offset QAM* merupakan *pulse shaping*, lalu pada bagian sinyal gq merupakan pergeseran pada bagian *quadrature*.



Gambar 3.12 *Offset QAM* Dengan *Pulse Shaping*

e) *Inverse Fast Fourier Transform (IFFT)*

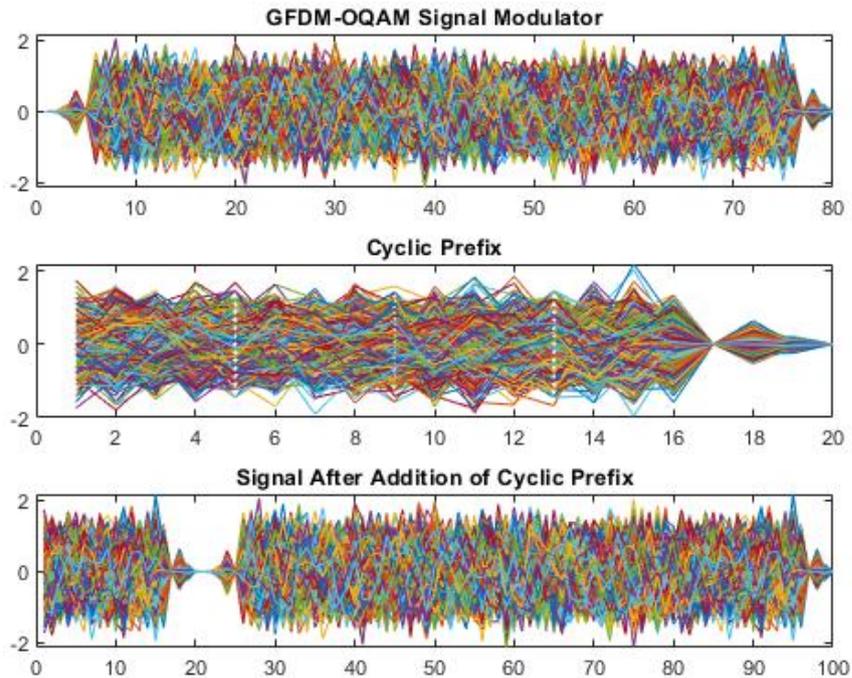
Setelah mendapatkan sinyal GFDM selanjutnya adalah proses IFFT, yaitu proses yang berfungsi melakukan proses untuk mengubah sinyal dari domain frekuensi ke domain waktu. Proses IFFT dari keluaran OQAM akan menghasilkan matriks A_i dan A_q . Matriks A_i dan A_q berukuran 80×80 yang merepresentasikan sinyal dari *subcarrier* per simbol pada *inphase* dan *quadrature* pada bagian riil dan imajiner. Pada Gambar 3.13 merupakan bentuk sinyal dari hasil IFFT yaitu sinyal *subcarrier* per simbol pada matriks A_i dan A_q .



Gambar 3.13 Sinyal *Subcarrier* Pada OQAM

5) Penambahan *Cyclic Prefix*

Proses terakhir pada bagian *transmitter* adalah penambahan *cyclic prefix* yang menghasilkan $\vec{\tilde{x}}$. CP bertujuan untuk mengatasi terjadinya penambahan *noise* berlebih yang diakibatkan oleh *pulse shaping*, sehingga *noise* tersebut dapat dikurangi sekecil mungkin. Pada sistem GFDM, CP diambil dari salinan akhir dari bit-bit suatu simbol dan kemudian diletakan di awal simbol. CP disisipkan pada setiap blok GFDM yang terdiri dari beberapa *subsymbol*. Blok GFDM *Modulator* (\vec{x}) menghasilkan matriks berukuran 80 x 1000. Besaran CP yang digunakan adalah 2% dari jumlah simbol yang digunakan yaitu 1000 simbol, sehingga menghasilkan panjang CP sebesar 20. Simbol GFDM yang akan disalin adalah 20 simbol blok GFDM terakhir, kemudian akan diletakkan pada awal simbol blok GFDM. Sehingga simbol GFDM yang awalnya berjumlah 80 akan menjadi 100 setelah dilakukan penambahan CP. Pada Gambar 3.14 adalah sinyal yang ditransmisikan setelah mengalami penambahan *cyclic prefix*.



Gambar 3.14 Sinyal Setelah Mengalami Penambahan CP

3.2.3.2 Kanal *Vehicle to Vehicle*

Kanal *vehicle to vehicle* merupakan sebuah pemodelan kanal yang ditentukan dengan model geometri. Pengirim (Tx) dan Penerima (Rx) bergerak dengan kecepatan konstan dan dikelilingi oleh *scatterer*. Diasumsikan bahwa pengirim dan penerima bergerak beserta *scatterer* bergerak yang mempunyai kecepatan acak. Pada kanal V2V arah propagasi dibedakan menjadi dua yaitu sinyal di kirim tanpa halangan LOS dan sinyal yang terkena *scatterer* bergerak. Pada penelitian ini kecepatan yang digunakan adalah 17 m/s, 22 m/s dan 30 m/s pada sisi pengirim dan penerima, untuk sisi *scatterer* sebesar 5 m/s. Jumlah *scatterer* yang digunakan adalah 8. Pada kanal V2V terjadi pergerakan pada sisi pengirim, penerima dan *scatterer* yang akan menghasilkan efek *Doppler*. Untuk mencari nilai *Doppler shift* pada kanal V2V menggunakan persamaan (2.26), sedangkan untuk mencari nilai *gain* pada kanal menggunakan persamaan (2.20). Untuk memvalidasi kanal tersebut dilakukan dengan proses distribusi dan autokorelasi, distribusi yang digunakan adalah distribusi normal (*Gaussian*) dengan menggunakan persamaan (2.27). Sedangkan untuk validasi autokorelasi menggunakan persamaan (3.1).

Berikut persamaan untuk mencari autokorelasi pada kanal V2V [4]:

$$r_{\mu\mu}(\tau) = \lim_{x \rightarrow \infty} \sum_{n=1}^N c_n^2 E \{ e^{2\pi f_n \tau} \} \quad (3.1)$$

di mana:

$$c_n = \sigma_0 \sqrt{2/N}$$

N = jumlah *scatterer* di sekitar area pengirim dan penerima

3.2.3.3 Bagian Receiver

Pada bagian *receiver*, dilakukan sinkronisasi waktu dan frekuensi, *output* yang dihasilkan adalah \tilde{y}_s . Untuk mencari nilai \tilde{y}_s atau Rx menggunakan persamaan sebagai berikut:

$$rx = x.h + w \quad (3.2)$$

di mana:

rx = sinyal yang diterima

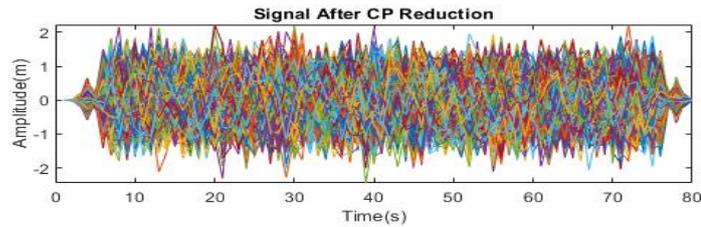
x = merupakan sampel waktu *transmit* $x[n]$, x merupakan hasil perhitungan dari persamaan (2.12)

h = *Gain* kanal V2V yang berbentuk bilangan kompleks, nilai h didapat dari persamaan (2.20)

w = *Noise*

1) Remove Cyclic Prefix

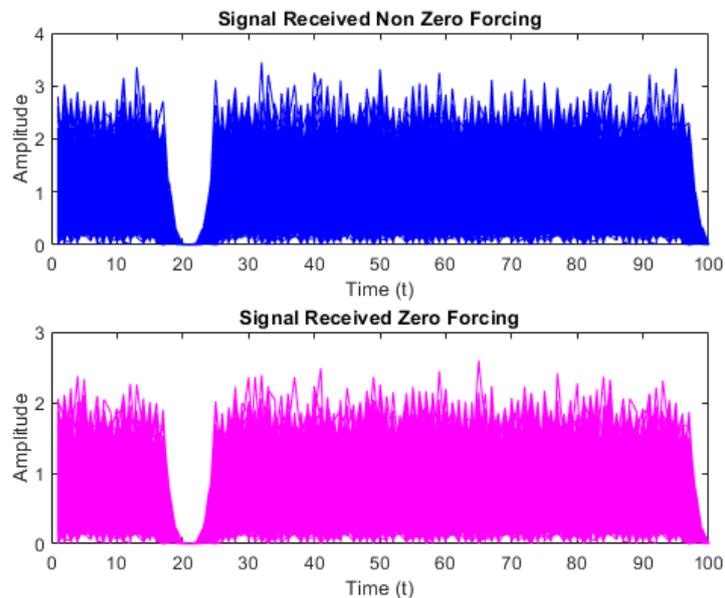
Proses awal pada sisi *receiver* yaitu *remove* atau menghilangkan *cyclic prefix* yang sebelumnya telah disisipkan pada setiap blok dari M simbol pada sisi *transmitter*. *Output* yang dihasilkan yaitu simbol yang hanya mengandung informasi saja yang tersisa dinotasikan dengan simbol \bar{y} . Pada bagian *transmitter* ukuran matriks GFDM yang telah ditambahkan CP sebesar 100×1000 , setelah dilakukan penghilangan CP ukuran matriks pada simbol berubah menjadi 80×1000 . Seperti yang terlihat pada Gambar 3.15 merupakan *output* sinyal yang sudah melewati proses pengurangan CP.



Gambar 3.15 Hasil Sinyal Setelah *Remove CP*

2) **Ekualisasi *Zero Forcing***

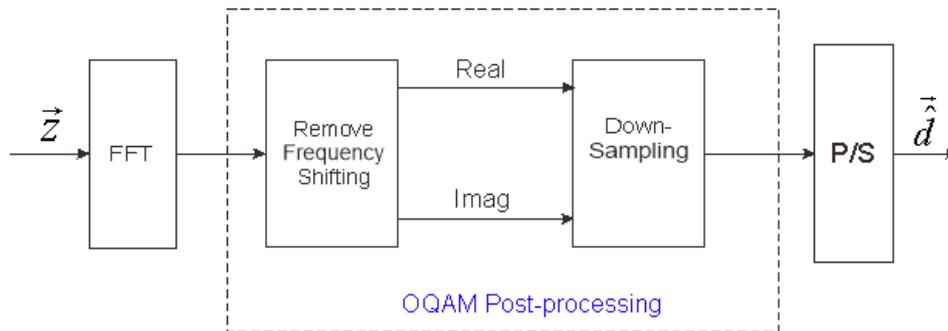
Proses pengolahan sinyal pada deteksi simbol bertujuan untuk mendapatkan nilai sinyal aslinya secara digital. Pada penelitian ini menggunakan ekualisasi *Zero Forcing* (ZF) karena ekualisasi ini memiliki kompleksitas paling rendah. Cara kerja ekualisasi *Zero Forcing* yaitu dengan cara menginverskan kanal H dan dikalikan dengan sinyal yang dikirim. Peneliti memilih ekualisasi ZF dikarenakan memiliki kompleksitas paling rendah diantara ekualisasi lainnya. *Output* dari ekualisasi ZF dinotasikan dengan simbol \tilde{z} dan menghasilkan matriks dengan ukuran sebesar 100 x 1000. Seperti yang terlihat pada Gambar 3.16 merupakan perbandingan sinyal yang menggunakan ekualisasi ZF dan tanpa ekualisasi ZF.



Gambar 3.16 Perbandingan Sinyal Non ZF dan Sinyal ZF

3) ***Demodulator GFDM***

Pada Gambar 3.17 berikut merupakan blok diagram *demodulator GFDM*.



Gambar 3.17 Blok Diagram *Demodulator* GFDM-OQAM

a) *Fast Fourier Transform*

Setelah dilakukan *remove cyclic prefix*, selanjutnya dilakukan proses FFT yaitu kembali dipisahkannya data informasi dengan sinyal *carrier*. Proses FFT juga disebut dengan proses demodulasi *multicarrier* yang merupakan kebalikan dari proses IFFT. Dalam proses ini domain waktu diubah kembali menjadi domain frekuensi. Proses FFT pada GFDM dilakukan dengan cara mengalikan bentuk *transpose* dari matriks A_i dan A_q hasil dari proses IFFT dengan sinyal yang ditransmisikan setelah pengurangan CP. Proses FFT menghasilkan ukuran matriks sebesar 8×10000 bernilai bilangan kompleks. Seperti yang terlihat pada Gambar 3.18 merupakan hasil simulasi FFT.

	1	2	3	4	5
1	-0.0238 - 0.0054i	1.3013 + 0.4877i	1.4120 + 0.4426i	1.1348 - 1.2050i	0.7442 + 1.5687i
2	0.0274 + 0.0336i	0.4982 + 0.4608i	0.5220 - 1.4285i	-1.2509 + 0.4328i	0.4440 + 1.4129i
3	-0.0170 - 0.0380i	1.1893 - 0.3862i	1.3528 - 0.4721i	0.3898 - 1.4407i	-1.4237 - 1.3600i
4	-0.0130 + 0.0265i	-1.3078 - 0.4471i	-0.4714 - 1.4972i	1.1573 - 0.3116i	0.1916 - 0.5798i
5	0.0124 - 0.0303i	-0.0618 - 0.0183i	-0.0136 + 0.0053i	0.1279 - 0.2252i	0.1176 + 0.0506i
6	0.0273 + 0.0393i	0.0644 - 0.0913i	-0.1378 - 0.0396i	-0.1347 - 0.0681i	0.0213 + 0.0392i
7	-0.0515 - 0.0196i	0.0040 + 0.0236i	-0.0351 + 0.0419i	-0.0439 + 0.1427i	-0.0610 - 0.2155i
8	0.0382 - 0.0060i	0.5775 + 1.2907i	0.4421 + 0.4884i	-0.2636 + 1.2698i	0.4888 - 0.4465i

Gambar 3.18 Nilai Simulasi FFT

b) *Remove Frequency Shifting*

Setelah dilakukan proses FFT, selanjutnya adalah *remove frequency shifting*, di mana sinyal yang diterima digeser kembali pada kondisi awal dengan modulasi *baseband* pada GFDM, dengan cara mengalikan matriks *inphase* dengan data setelah penghapusan CP lalu diambil bagian riil. Kemudian mengalikan matriks

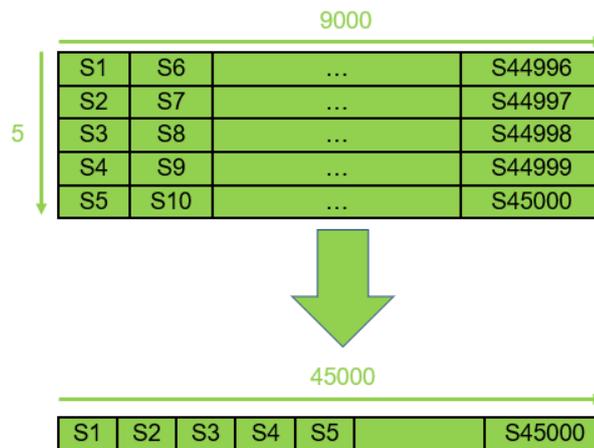
quadrature dengan data setelah penghapusan CP, lalu diambil bagian imajineranya. Setelah itu hasil perkalian matriks *inphase* dan *quadrature* digabung dan menghasilkan sampel per simbol. Proses ini menghasilkan matriks dengan ukuran sebesar 8×10000 dengan nilai bilangan kompleks.

c) Down Sampling

Dari hasil proses FFT berbentuk paralel, namun tidak semua baris pada bentuk paralel tersebut dipakai untuk diproses lebih lanjut. Hal ini disesuaikan dengan jumlah *subcarrier* yang digunakan, pada penelitian ini digunakan 5 *subcarrier*. Sehingga ukuran matriks yang digunakan adalah ukuran 5×9000 dengan nilai bilangan kompleks.

d) Parallel to Serial

Hasil dari proses *down-sampling* dengan ukuran matriks 5×9000 dengan melakukan proses konversi dari *parallel* menjadi *serial* dengan menggunakan fungsi *reshape* pada matlab, ukuran matriks yang dihasilkan sebesar 1×45000 , seperti yang terlihat pada Gambar 3.19 merupakan hasil konversi *parallel to serial*.

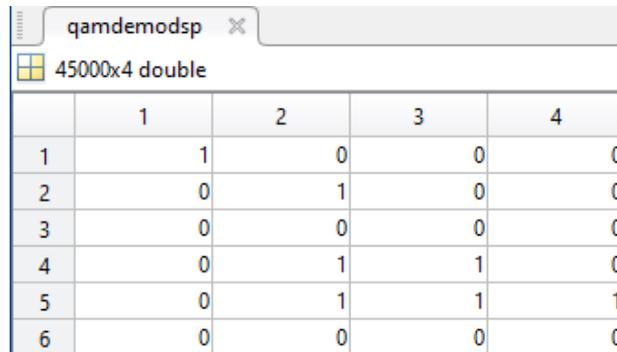


Gambar 3.19 Konversi *Parallel to Serial*

4) Demapping 16-QAM

Setiap data sinyal yang diterima digeser kembali pada kondisi awal dengan modulasi 16-QAM yang selanjutnya dilakukan *demapping* atau demodulasi. Tujuan dari demodulasi adalah untuk mengubah deretan simbol GFDM menjadi

bit-bit informasi kembali. Bit-bit informasi dinotasikan dengan simbol ($\vec{\hat{b}}_c$). Pada Gambar 3.20 terlihat hasil bit-bit informasi yang dihasilkan dari proses *demapping*.

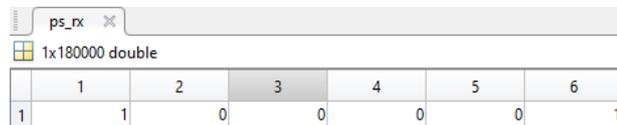


	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	0	0
4	0	1	1	0
5	0	1	1	1
6	0	0	0	0

Gambar 3.20 Hasil Nilai *Demapping*

5) *Decoder*

Setelah melewati proses *demapping*, selanjutnya dilakukan proses *decoder* yaitu kebalikan dari proses *encoder*, di mana bentuk bit yang dihasilkan *demapping* masih dalam berbentuk paralel. Selanjutnya diubah menjadi bentuk serial seperti bentuk awal dengan bentuk bit biner yang menghasilkan bit keluaran. Keluaran yang dihasilkan berukuran 1×180000 dan dinotasikan dengan simbol $\vec{\hat{b}}$. Seperti yang terlihat pada Gambar 3.21 merupakan hasil nilai *decoder*.



	1	2	3	4	5	6
1	1	0	0	0	0	1

Gambar 3.21 Hasil Nilai *Decoder*

6) *Bit Output*

Hasil dari *decoder* yang disimbolkan dengan $\vec{\hat{b}}$ dijadikan sebagai bit *output* pada sistem ini. Proses selanjutnya bit-bit biner yang diterima ini akan dibandingkan dengan bit-bit awal yang ada pada sisi *transmitter*. Keluaran bit informasi ini digunakan untuk mengetahui nilai *Bit Error Rate* (BER) atau kesalahan bit yang diterima pada sisi *receiver* dibandingkan dengan bit informasi awal yang dibangkitkan pada sisi *transmitter*. Seperti yang terlihat pada Gambar 3.22 dan Gambar 3.23 merupakan bit *output* tanpa ekualisasi ZF dan menggunakan ekualisasi ZF.

ps_rx						
1x180000 double						
	61	62	63	64	65	66
1	0	0	1	0	1	1

Gambar 3.22 Hasil Bit *Output Non Zero Forcing*

ps_rx2						
1x180000 double						
	61	62	63	64	65	66
1	0	0	1	0	1	0

Gambar 3.23 Hasil Bit *Output Zero Forcing*

7) Simulasi BER

Pada penelitian ini, kinerja dari sistem *multicarrier* GFDM-OQAM menggunakan kanal V2V dengan *scatterer* bergerak dengan menggunakan ekualisasi ZF dan tanpa ekualisasi ZF. Hasilnya dibandingkan dengan BER teori yang menggunakan persamaan (2.45).