

BAB 3

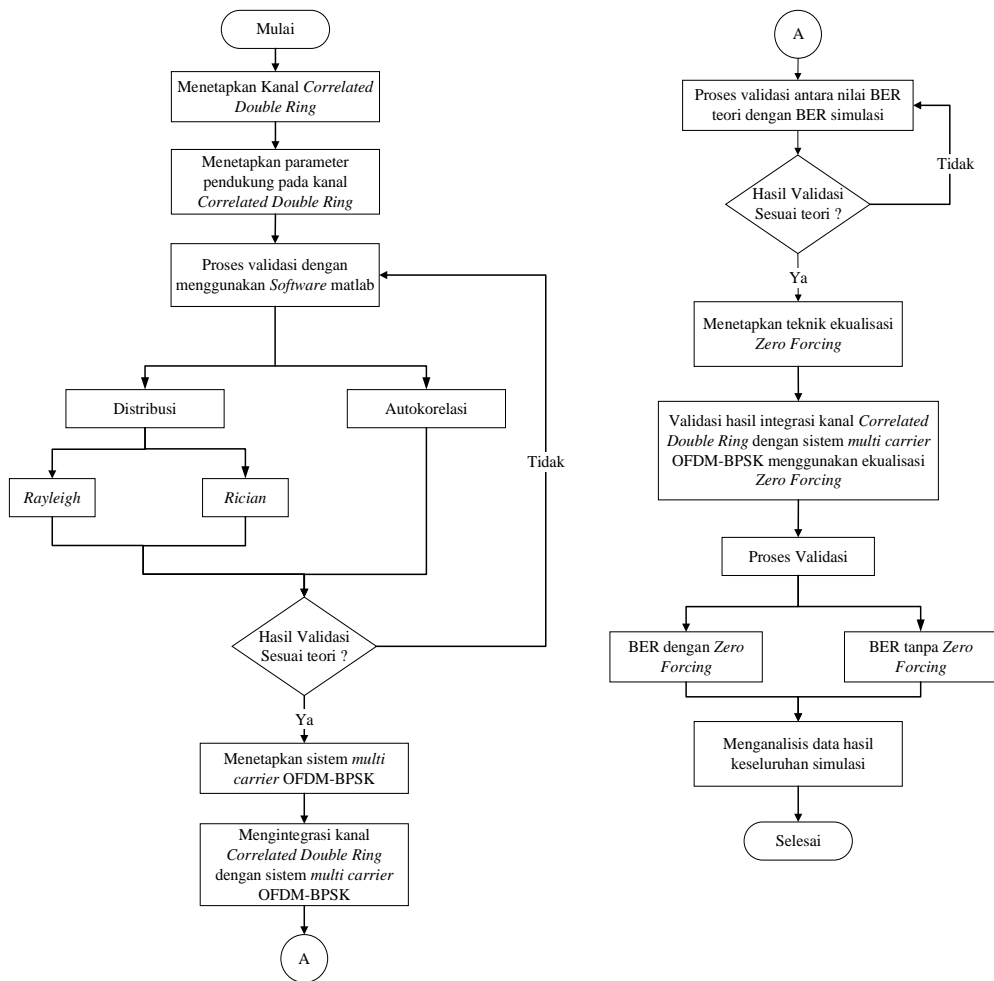
METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

Pembuatan program simulasi ini digunakan untuk mengetahui hasil integrasi model kanal *Correlated Double Ring* dengan sistem *multicarrier* OFDM dan teknik ekualisasi *Zero Forcing* untuk memitigasi efek *Doppler* melalui parameter BER. Modulasi yang digunakan yaitu modulasi BPSK. Alat yang digunakan pada penelitian ini adalah *software* MATLAB R2015a.

3.2 ALUR PENELITIAN

Rancangan penyelesaian skripsi, mengenai Teknik Ekualisasi *Zero Forcing* pada Sistem *Multicarrier* OFDM Menggunakan Kanal *Correlated Double Ring* dijabarkan dalam diagram alir (*flowchart*) yang ditunjukkan pada Gambar 3.1. Penelitian ini dilakukan dalam beberapa tahap yaitu *study literature*, tahap perancangan, tahap pembuatan simulasi, tahap pengujian dan tahap analisis dari hasil pengujian simulasi. Pertama dimulai dengan menentukan model kanal yang akan digunakan. Kanal yang digunakan pada simulasi ini yaitu model kanal *Correlated Double Ring* (CDR). Selanjutnya melakukan validasi model kanal CDR melalui validasi distribusi dan autokorelasi *Rayleigh* dan *Rician*. Setelah menentukan model kanal, selanjutnya yaitu menentukan sistem *multicarrier* yang digunakan. Pada simulasi ini menggunakan *multicarrier* OFDM. Modulasi yang digunakan yaitu BPSK. Langkah selanjutnya yaitu mengintegrasikan kanal CDR dengan sistem *multicarrier* OFDM yang dilakukan dengan cara mengalikan *output* dari blok *transmitter* OFDM dengan kanal CDR ditambah dengan *noise* AWGN, kemudian divalidasi dengan parameter BER. Setelah itu, menentukan ekualisasi pada sisi penerima. Ekualisasi yang digunakan yaitu *Zero Forcing*. Selanjutnya mengintegrasikan kanal *correlated double ring* dengan sistem *multicarrier* OFDM dan ekualisasi *Zero Forcing* dengan parameter BER. Langkah terakhir validasi BER dengan *Zero Forcing* dan BER tanpa *Zero Forcing* secara teori dan simulasi. Apabila belum sesuai teori maka harus dilakukan perbaikan pada program. Jika sudah sesuai dengan teori, dilanjutkan dengan analisis.



Gambar 3. 1 Flowchart rancangan Simulasi Program

3.3 PARAMETER SIMULASI

Adapun parameter yang digunakan pada simulasi ini dapat dilihat pada Tabel 3.1 dan Tabel 3.2 berikut.

Tabel 3. 1 Parameter *multicarrier* OFDM

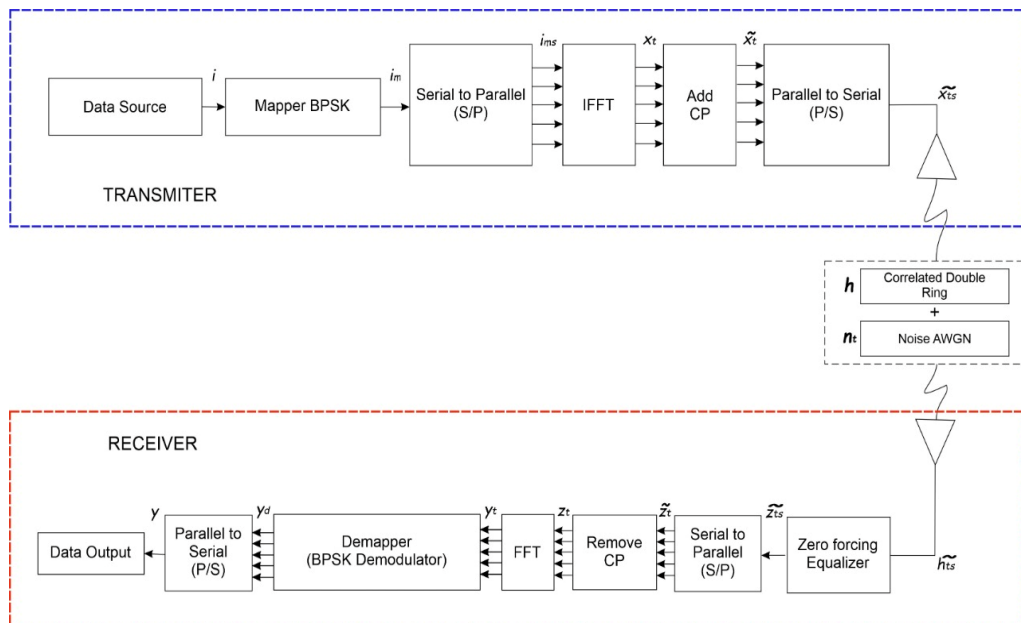
Simbol	Parameter	Nilai
	Jenis modulasi	BPSK
m	Jumlah level modulasi	1
h	kanal	<i>Correlated double ring</i>
M,N	Jumlah <i>scattering</i>	8
	Ekualisasi	<i>Zero forcing</i>
i	Jumlah data masukan	52000 bit
nSym	Jumlah simbol	1000
f_c	Frekuensi pembawa	$5,8 \times 10^9$ Hz
Tb	Jumlah FFT	64

Tabel 3. 2 Parameter Kanal *Correlated Double Ring*

Simbol	Parameter	Nilai
$V1 = V2$	Kecepatan pada sisi TX dan RX	5 m/s; 20 m/s dan 55 m/s
c	Kecepatan cahaya	3×10^8
M, N	Jumlah penghambur pada sisi pengirim dan penerima	8
K	Perbandingan daya yang diterima dan daya yang dipantulkan	8
θ_n, ψ_m	Sudut kedatangan dan sudut pengiriman	Bernilai bilangan bulat acak diantara $(-180^\circ) - 180^\circ$ dengan distribusi seragam
θ_{send}	Sudut antara arah pergerakan kendaraan pengirim dengan garis horisontal	45°
θ_{diff}	Sudut antara $V1$ dan $V2$	60°
t	<i>Time Delay</i>	0:100 s

3.4 Pemodelan Sistem OFDM

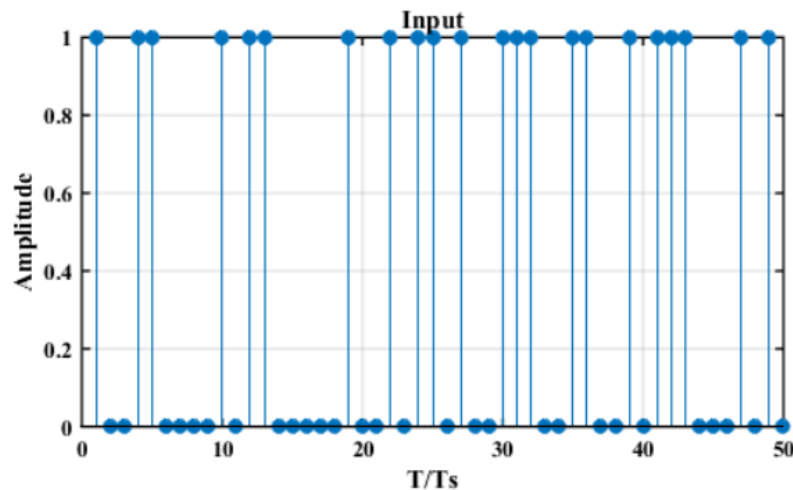
Proses pemodelan sistem OFDM dapat dijelaskan pada Gambar 3.2 berikut ini



Gambar 3. 2 Pemodelan Sistem OFDM

3.4.1 Data Masukan

Data masukan pada blok diagram rancangan pemodelan merupakan suatu pembangkit data (*data generator*). Data generator ini fungsinya untuk membangkitkan data biner sebagai data informasi yang akan dikirim. Pada penelitian ini bit informasi (*i*) yang akan dibangkitkan berjumlah 52000 bit. Data yang dibangkitkan nilainya sesuai dengan level modulasi yang digunakan. Bit informasi dibangkitkan secara acak atau *random* dengan menggunakan fungsi *randsrc* pada *software* MATLAB R2015a, sehingga dihasilkan bit 1 dan bit 0 dengan pola acak dan memiliki peluang kemunculan yang sama. Sinyal informasi bit yang dibangkitkan dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Sinyal *Input*

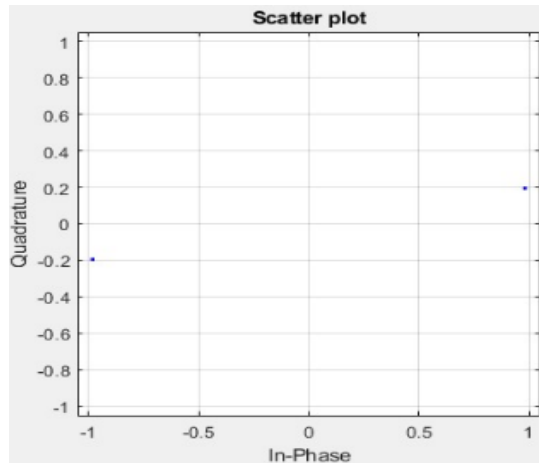
1	2	3	4	5	6
1	0	0	1	1	0

Gambar 3. 4 *Bit Biner* Sinyal *Input*

3.4.2 Mapper BPSK

Pada blok *mapper*, bit informasi (*i*) yang telah dibangkitkan akan dikirim dan dikodekan menggunakan modulasi BPS). Pada BPSK, fase dari frekuensi pembawa di ubah antara dua nilai yang menyatakan keadaan biner 1 dan 0. Dalam hal ini, fase dari frekuensi pembawa satu dengan yang lain berbeda fase sebesar π radian atau 180° . Diagram konstelasi modulasi BPSK dapat dilihat pada Gambar 3.5.

Berdasarkan persamaan (2.32) dan persamaan (2.33), maka *output* (im) modulasi BPSK akan memetakan 1 bit menjadi 1 simbol yang bernilai +1 atau -1 seperti yang tertera pada Gambar 3.6.



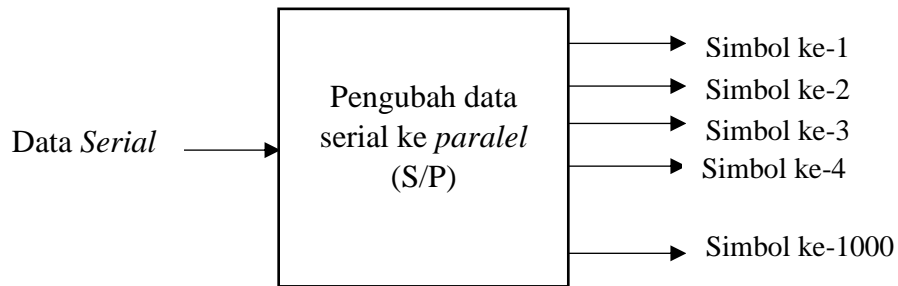
Gambar 3. 5 Diagram Konstelasi BPSK

1	2	3	4	5	6
1	-1	-1	1	1	-1

Gambar 3. 6 Mapper BPSK

3.4.3 Serial to Parallel (S/P)

Blok *serial to parallel* berfungsi untuk mengubah aliran data yang terdiri dari satu baris menjadi beberapa baris dan beberapa kolom. Proses *serial to parallel* dapat dilakukan setelah bit informasi dimodulasi menjadi simbol data. Simbol data (im) yang berjumlah 1x52000 dikirim secara serial kemudian diubah menjadi bentuk *parallel* berdasarkan jumlah *subcarrier* yang digunakan. Jumlah *subcarrier* yang digunakan yaitu 52 sehingga jumlah simbol data pada setiap *subcarrier* adalah 1000 simbol data. Maka *output* dari (ims) berupa matriks 1000x52. Pada simulasi ini proses perubahan *serial to parallel* dilakukan dengan menggunakan fungsi *reshape* pada Matlab.



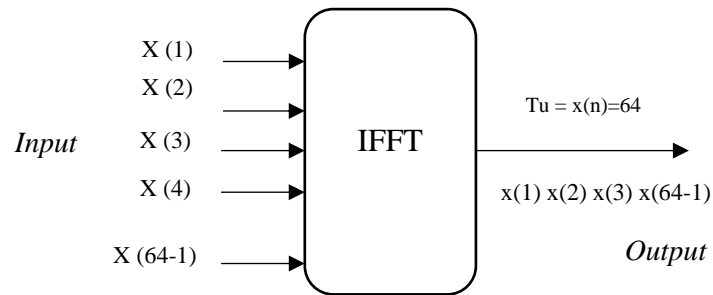
Gambar 3. 7 Pengubah data serial ke *parallel* [5]

1000x52 double											
	1	2	3	4	5	6	7	8	9	10	11
3	1	1	1	1	1	1	1	1	1	1	1
4	-1	-1	1	1	1	1	1	-1	-1	-1	1
5	-1	1	1	-1	-1	-1	1	1	1	1	1
6	1	1	-1	-1	-1	1	1	1	1	-1	-1
7	-1	1	1	1	1	-1	1	1	-1	-1	1
8	-1	-1	1	1	1	-1	-1	-1	1	-1	-1
9	1	1	1	1	-1	-1	1	-1	1	-1	-1
10	1	1	1	-1	-1	1	1	1	-1	-1	-1

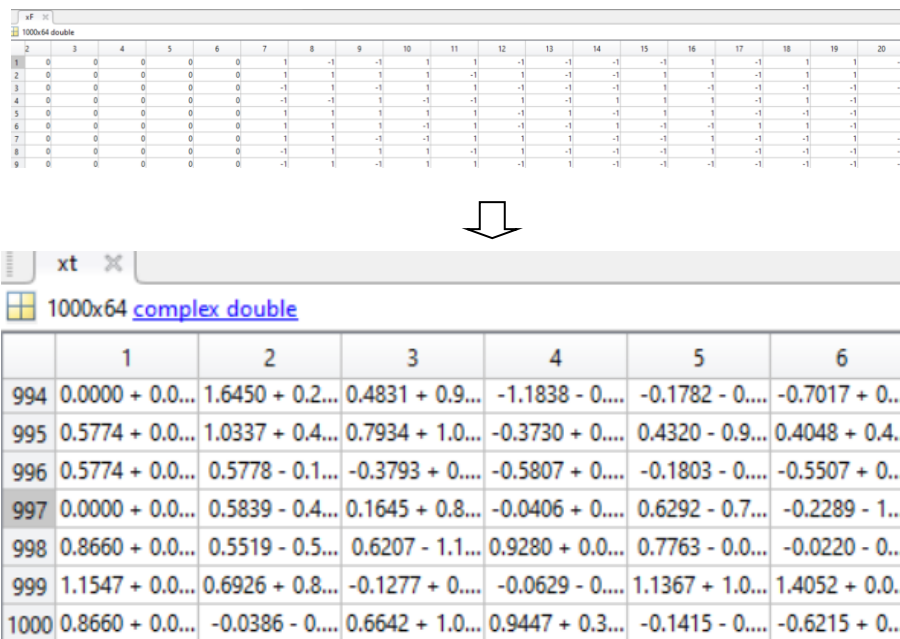
Gambar 3. 8 *Output* dari S/P di sisi *Transmitter*

3.4.4 Proses *Inverse Fast Fourier Transform* (IFFT)

IFFT merupakan proses untuk menghasilkan simbol-simbol OFDM pada sisi *transmitter* dengan frekuensi dari setiap informasinya akan dibuat saling tegak lurus (*Orthogonal*) dan subkanal dapat saling *overlapping* tanpa menimbulkan *interference*. Proses pada IFFT sering disebut juga proses modulasi pada OFDM. IFFT mengubah spektrum yaitu amplitudo dan fasa dari setiap sinyal informasi ke bentuk sinyal dalam domain waktu. Data yang dibutuhkan dalam proses IFFT yaitu dalam bentuk data *parallel*, sehingga dalam bentuk matriks baris dan kolom seperti yang ditunjukkan pada Gambar 3.8. Adapun jumlah N yang digunakan pada IFFT adalah 64. Elemen dari baris merupakan data dari *subcarrier* dan kolom merupakan data dari simbol yang akan dibangkitkan, sehingga keluaran dari IFFT (x_t) yaitu matriks 1000x64. Pada simulasi ini proses IFFT dilakukan dengan menggunakan fungsi “*ifft*” pada Matlab.



Gambar 3. 9 Bagan IFFT



Gambar 3. 10 Hasil keluaran IFFT

3.4.5 Penambahan *Cyclic prefix* (CP)

Pada sistem OFDM, CP digunakan untuk mengatasi efek dari *Intersymbol Interference* (ISI) akibat kanal *multipath fading*. Simbol yang telah dilengkapi CP akan mampu dilakukan *recovery* dengan baik oleh sisi penerima walaupun terkena interferensi *fading* dari kanal yang cukup besar. CP mengacu pada awalan simbol OFDM yang merupakan pengulangan bagian akhir dari simbol OFDM. Jumlah CP yang digunakan adalah $\frac{1}{4}$ dari jumlah simbol OFDM pada setiap *subcarrier*. Jumlah simbol OFDM pada setiap *subcarrier* berjumlah 64 sehingga jumlah CP adalah 16. Setelah diketahui jumlah CP yang akan digunakan maka akan dilakukan penyalinan pada simbol OFDM pada setiap *subcarrier*. 16 simbol OFDM terakhir disalin dan

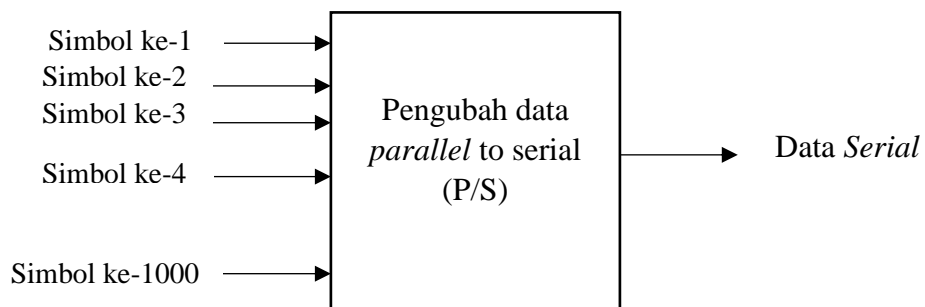
diletakkan pada awal simbol OFDM. *Output* dari IFFT (x_t) yaitu 1000×64 sehingga simbol OFDM setelah ditambahkan CP berjumlah 1000×80 . *Output* dari penambahan CP pada pemodelan sistem dinyatakan dengan \widetilde{X}_t .

	1	2	3	4	5	6
993	1.1547 + 0.2887i	-0.7940 - 0.0009i	-0.2439 - 1.2856i	0.1058 - 0.8604i	-1.0024 + 1.4732i	0.9111 + 1.1521i
994	-0.2887 + 0.0000i	-0.9673 - 0.7344i	-0.0499 - 1.7625i	0.5485 - 0.5365i	-0.5554 + 0.2926i	-0.7616 + 0.0073i
995	-1.1547 - 0.8660i	-0.4183 + 0.5743i	-0.1285 + 0.6935i	-0.2771 - 0.2772i	0.0574 - 0.3793i	-0.3250 - 0.9073i
996	1.1547 - 0.8660i	-0.3811 - 0.4779i	0.8330 + 0.1300i	1.1140 - 2.7971i	0.6698 - 0.5554i	-0.0470 + 1.1103i
997	0.5774 - 0.8660i	-0.3876 + 0.6373i	0.6363 + 0.3002i	-0.0345 - 0.0947i	-0.9417 - 0.4470i	-0.4442 + 0.4029i
998	0.0000 - 0.5774i	-0.8370 - 0.7329i	0.1113 + 0.5293i	0.9984 - 0.4076i	-0.1343 - 0.6857i	0.3677 + 0.1093i
999	-0.8660 + 0.0000i	0.4635 - 0.6464i	-0.2319 - 0.7616i	-0.3366 + 0.0280i	0.0388 + 0.1899i	-0.4213 + 0.8929i
1000	0.0000 + 1.1547i	0.3200 - 0.0791i	0.3082 - 0.2449i	0.2104 + 1.1027i	0.5187 - 1.1514i	0.6814 - 1.9546i

Gambar 3. 11 *Output add CP*

3.4.6 *Parallel to serial (P/S)*

Setelah melakukan penambahan CP (\widetilde{X}_t) pada sinyal *parallel* maka sinyal akan dikirimkan oleh *transmitter*. Namun pada sistem OFDM sinyal yang akan dikirim harus berbentuk sinyal tunggal sehingga harus dilakukan konversi dari bentuk *parallel* menjadi bentuk serial. *Output* dari CP yang semula terdiri dari 1000×80 diubah menjadi 1×80000 dengan menggunakan fungsi *reshape* pada Matlab.



Gambar 3. 12 Pengubah data *parallel* ke serial

	1	2	3	4	5	6
1	1.1547 + 1.1547i	-1.5381 - 0.4963i	-0.9376 - 0.6415i	-0.3936 - 0.5482i	-0.9636 + 0.6771i	-0.4921 + 0.3525i

Gambar 3. 13 *Output* dari P/S disisi *Transmitter*

3.4.7 Pemodelan kanal *Correlated Double Ring*

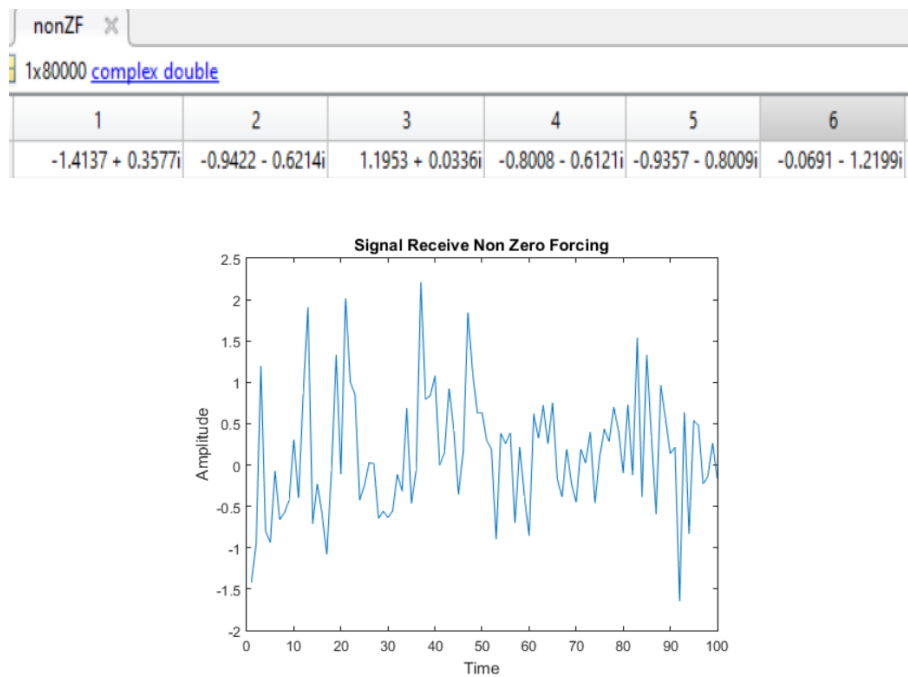
Langkah awal perancangan pemodelan kanal *Correlated Double Ring* dimulai dengan menentukan parameter pendukung kanal yang sudah ditentukan nilainya seperti yang terlihat pada Tabel 3.2.

Proses yang kedua yaitu proses simulasi pemodelan kanal *Correlated Double Ring* yang terdiri dari proses simulasi *Rayleigh Fading* dan *Rician Fading* seperti pada Gambar 2.1. *Rayleigh Fading condition* merupakan pemodelan kanal tanpa komponen LOS, sinyal dikirim dengan terkena kondisi *multipath fading*. Sinyal yang diterima pada kondisi *Rayleigh Fading* dinyatakan pada persamaan (2.5) yang dinotasikan dengan sinyalYT. Untuk mendapatkan nilai dari sinyalYT mengacu pada persamaan (2.6) dan persamaan (2.7). Selanjutnya untuk pemodelan kanal pada *Rician Fading channel* ini menyatakan pemodelan dimana kondisi sinyal yang diterima mempertimbangkan objek yang dipantulkan (sinyalYT) ditambah dengan satu komponen yang dominan yaitu berasal dari komponen *Line of Sight* (LOS). Sinyal *Rician Fading* dinotasikan dengan sinyalZt seperti pada persamaan (2.9). Untuk mendapatkan nilai dari sinyalZt mengacu pada persamaan (2.1), persamaan (2.2), persamaan (2.3), persamaan (2.4) dan persamaan (2.5).

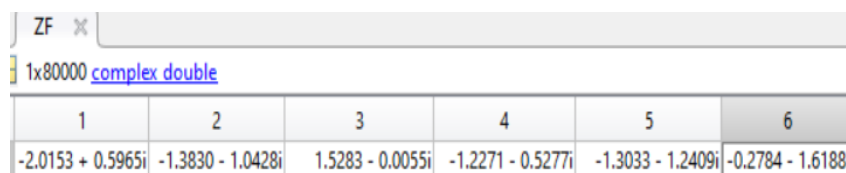
Tahap terakhir yaitu proses validasi. Proses validasi dilakukan dengan 2 tahap yaitu validasi distribusi dan validasi autokorelasi untuk *Rayleigh* dan *Rician Fading channel*. Validasi distribusi *Rayleigh* dilakukan dengan membandingkan sinyalYt pada persamaan (2.5) yang sudah dinormalisasi dengan *output* dari PDF pada persamaan (2.20), sedangkan validasi *Rician* dilakukan dengan cara membandingkan sinyalZt pada persamaan (2.9) yang sudah dinormalisasi dengan *output* dari PDF pada persamaan (2.23). Proses validasi distribusi dikatakan selesai jika validasi secara simulasi sudah sesuai dengan teori. Validasi autokorelasi *Rayleigh* dan *Rician* dilakukan dengan cara menambahkan notasi “tha” dan *delay* pada nilai *real* dari sinyal hasil distribusi *Rayleigh* (sinyalYt) pada persamaan (2.5) dan nilai *real* dari sinyal hasil distribusi *Rician* (sinyalZt) pada persamaan (2.9). “tha” merupakan notasi dari parameter fungsi *Bessel* awal. Hasil validasi autokorelasi *Rayleigh* dinotasikan dengan notasi “RYY” sedangkan hasil autokorelasi *Rician* dinotasikan dengan notasi “RZZ”.

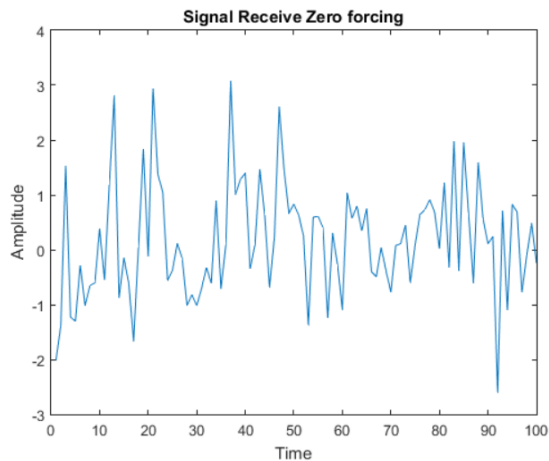
3.4.8 Proses Ekualisasi

Proses ekualisasi merupakan proses mencari respon agar kanal dapat diatasi. Proses ekualisasi dilakukan dengan mengolah sinyal secara digital untuk mendapatkan kembali sinyal data yang dikirimkan. Pada simulasi ini menggunakan proses ekualisasi *Zero Forcing* (ZF) karena ZF memiliki kompleksitas yang paling rendah jika dibandingkan dengan ekualisasi yang lain. Ekualisasi ZF dilakukan dengan cara mengalikan *invers* kanal (h) dengan *receive* ($\widehat{h}ts$). *Output* dari ekualisasi ZF pada pemodelan sistem dinyatakan dengan ($\widehat{Z}ts$) yang berukuran 1×80000 .



Gambar 3. 14 *Output Receive NonZero Forcing*

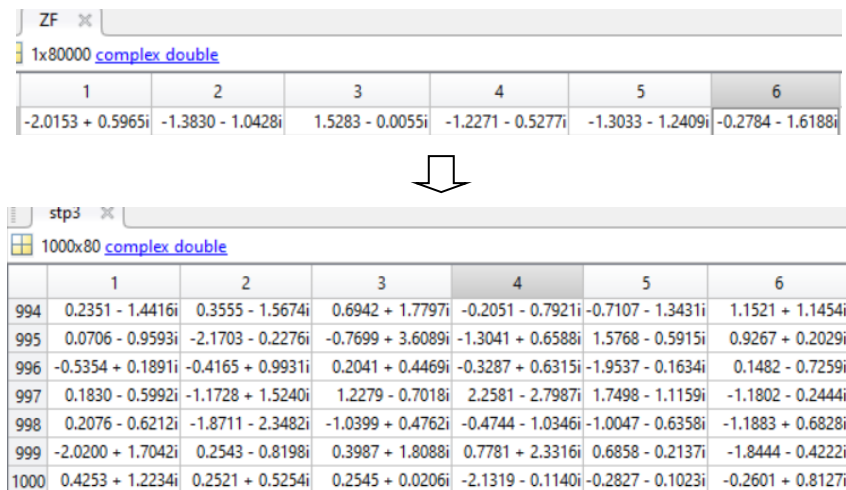




Gambar 3. 15 *Output Receive Zero Forcing*

3.4.9 Serial to Parallel (S/P)

Pada blok ini, hasil keluaran dari blok diagram *zero forcing* ($\widetilde{Z}ts$) yang awalnya dalam bentuk serial 1×80000 dikonversi kembali ke dalam bentuk *parallel* sehingga menghasilkan keluaran 1000×80 . Untuk merubah data dari bentuk serial ke *parallel* digunakan fungsi *reshape* pada Matlab. *Output S/P* pada sisi *receiver* dari pemodelan sistem dinyatakan dengan notasi ($\widetilde{Z}t$).



Gambar 3. 16 *Output Serial to Parallel ZF pada sisi Receiver*

3.4.10 Menghilangkan CP

Pada sisi penerima, CP akan dibuang sehingga akan didapatkan kembali data informasi yang sebenarnya. Penghapusan CP dilakukan dengan cara menghilangkan 1/4 dari jumlah simbol OFDM pada setiap *subcarrier* yaitu sebesar 16. Sinyal yang telah ditambah CP berjumlah 80 maka setelah proses pelepasan CP sinyal akan kembali seperti semula yaitu sebanyak 64, seperti yang terlihat pada Gambar 3.17. *Output remove CP* pada pemodelan sistem dinyatakan dengan (Zt).

	1	2	3	4	5	6
994	0.2351 - 1.4416i	0.3555 - 1.5674i	0.6942 + 1.7797i	-0.2051 - 0.7921i	-0.7107 - 1.3431i	1.1521 + 1.1454i
995	0.0706 - 0.9593i	-2.1703 - 0.2276i	-0.7699 + 3.6089i	-1.3041 + 0.6588i	1.5768 - 0.5915i	0.9267 + 0.2029i
996	-0.5354 + 0.1891i	-0.4165 + 0.9931i	0.2041 + 0.4469i	-0.3287 + 0.6315i	-1.9537 - 0.1634i	0.1482 - 0.7259i
997	0.1830 - 0.5992i	-1.1728 + 1.5240i	1.2279 - 0.7018i	2.2581 - 2.7987i	1.7498 - 1.1159i	-1.1802 - 0.2444i
998	0.2076 - 0.6212i	-1.8711 - 2.3482i	-1.0399 + 0.4762i	-0.4744 - 1.0346i	-1.0047 - 0.6358i	-1.1883 + 0.6828i
999	-2.0200 + 1.7042i	0.2543 - 0.8198i	0.3987 + 1.8088i	0.7781 + 2.3316i	0.6858 - 0.2137i	-1.8444 - 0.4222i
1000	0.4253 + 1.2234i	0.2521 + 0.5254i	0.2545 + 0.0206i	-2.1319 - 0.1140i	-0.2827 - 0.1023i	-0.2601 + 0.8127i

	1	2	3	4	5	6
994	-0.6452 + 0.1031i	-0.2215 + 2.1427i	-0.5062 + 0.8920i	-1.3644 + 0.0478i	-0.7819 - 0.0179i	-0.5204 - 0.6010i
995	3.5801 - 1.6284i	-0.2515 - 2.0113i	-0.2847 - 0.3251i	0.7234 - 0.9056i	1.3529 - 0.7118i	1.3579 - 0.5591i
996	0.4610 + 0.2344i	1.6359 + 0.3179i	1.0227 - 2.4713i	0.8364 - 1.4854i	0.1009 - 0.6116i	-1.3396 + 0.9144i
997	-0.2471 + 1.6155i	0.3668 - 0.3313i	-0.4695 - 0.5673i	0.2538 - 1.1350i	-2.5269 - 1.0655i	0.9652 + 1.4057i
998	-1.8240 - 1.4205i	-2.8681 - 0.8423i	-0.4521 + 0.7963i	-0.8807 - 0.8887i	0.7057 - 0.8508i	1.5854 + 0.5286i
999	0.6387 - 0.6142i	-0.2050 - 2.8843i	-0.8631 - 1.2112i	-1.2188 - 1.5502i	-0.3493 + 0.3414i	-2.0217 - 0.4548i
1000	0.1922 + 0.0260i	-0.2704 + 1.5162i	-1.2055 + 1.1031i	1.2998 + 1.4053i	4.2533 + 1.6829i	0.3300 - 1.3987i

Gambar 3. 17 Ouput Removing CP

3.4.11 Proses Fast Fourier Transform (FFT)

FFT merupakan kebalikan dari proses IFFT yaitu merubah dari domain waktu ke domain frekuensi. Proses FFT bertujuan untuk memisahkan antara frekuensi *carrier* dengan simbol OFDM yang diterima pada sisi *receiver* sebelum didemodulasi dan diubah kembali kedalam bentuk bit informasi. Proses FFT dapat dibangkitkan pada Matlab dengan menggunakan fungsi “fft”. *Output* FFT dapat dilihat pada Gambar 3.18 yang dinyatakan dengan notasi (yt) berdasarkan Gambar 3.2.

yF2						
1000x64 complex double						
	1	2	3	4	5	6
994	0.1091 - 1.1212i	-1.5794 - 0.5169i	0.3043 + 1.0520i	-0.8064 - 0.6311i	-1.0581 - 0.2944i	-0.5671 - 0.0061i
995	1.0811 - 0.8129i	0.5841 - 0.1979i	1.0233 + 0.0471i	0.7288 + 1.0584i	-0.1506 + 0.5198i	-0.0638 - 0.1733i
996	-0.6440 - 0.4420i	0.8569 - 0.4468i	0.4853 + 0.5359i	0.0743 + 1.0077i	-0.2144 - 0.5181i	0.8096 + 1.2553i
997	0.3158 - 0.3517i	-0.1617 + 1.5921i	0.8977 + 0.4999i	0.5453 - 0.4345i	1.2554 - 0.5237i	0.3961 - 0.7015i
998	1.6255 + 0.3095i	0.6728 + 0.2322i	-1.2155 - 1.0723i	0.6244 - 1.0639i	-0.1164 - 0.0814i	-0.3496 + 0.6484i
999	0.0174 - 0.9817i	1.1879 + 0.2237i	-0.3095 + 0.4042i	0.0727 + 0.9619i	0.2377 - 0.1004i	-1.1494 + 0.4357i
1000	0.2525 - 0.2341i	0.3335 + 0.8242i	1.5889 - 0.2104i	-0.5630 + 1.8110i	0.4909 + 0.7443i	-0.2241 + 1.1036i



yMod2						
1000x52 complex double						
	1	2	3	4	5	6
994	-0.8682 - 0.7441i	-0.2190 + 1.0632i	0.4642 + 0.3196i	1.0708 + 0.8160i	1.1104 + 0.4922i	1.7459 - 0.0935i
995	0.0383 + 1.0365i	1.3048 - 0.5500i	0.1988 - 0.0586i	-1.6338 - 1.2568i	-1.0449 - 0.8238i	-0.3761 - 0.1423i
996	1.0952 + 0.8328i	1.2549 - 0.5402i	-0.5514 + 0.0728i	-1.1298 + 0.3253i	-0.7606 + 0.0198i	0.5863 - 1.5828i
997	-0.6965 + 1.3849i	-0.2155 - 0.0433i	1.0196 - 0.4203i	1.1233 - 0.0952i	1.6079 - 0.4534i	-0.7888 + 0.15...
998	0.2980 - 0.5691i	-1.6720 - 0.5296i	1.2445 + 0.0337i	0.9480 + 0.8224i	0.4033 + 0.6098i	-1.7713 - 0.2140i
999	1.2950 - 0.3888i	0.1986 + 0.5514i	1.4676 + 0.8746i	2.1581 + 0.6301i	-0.8469 - 0.6548i	-0.0633 + 0.79...
1000	0.6287 + 0.3635i	1.4381 + 0.0362i	1.7187 - 0.3926i	-1.3747 + 0.6045i	-0.5371 - 0.2786i	0.6159 + 0.1963i

Gambar 3. 18 Output FFT

3.4.12 Demodulasi

Pada proses demodulasi, sinyal *output* dari FFT (yt) akan dipisahkan antara sinyal pembawa dan sinyal data, kemudian gelombang data yang terdeteksi akan dikembalikan dalam bentuk deret data *binary* seperti yang terlihat pada Gambar 3.19.

yMod2						
1000x52 complex double						
	1	2	3	4	5	6
994	-0.8682 - 0.7441i	-0.2190 + 1.0632i	0.4642 + 0.3196i	1.0708 + 0.8160i	1.1104 + 0.4922i	1.7459 - 0.0935i
995	0.0383 + 1.0365i	1.3048 - 0.5500i	0.1988 - 0.0586i	-1.6338 - 1.2568i	-1.0449 - 0.8238i	-0.3761 - 0.1423i
996	1.0952 + 0.8328i	1.2549 - 0.5402i	-0.5514 + 0.0728i	-1.1298 + 0.3253i	-0.7606 + 0.0198i	0.5863 - 1.5828i
997	-0.6965 + 1.3849i	-0.2155 - 0.0433i	1.0196 - 0.4203i	1.1233 - 0.0952i	1.6079 - 0.4534i	-0.7888 + 0.15...
998	0.2980 - 0.5691i	-1.6720 - 0.5296i	1.2445 + 0.0337i	0.9480 + 0.8224i	0.4033 + 0.6098i	-1.7713 - 0.2140i
999	1.2950 - 0.3888i	0.1986 + 0.5514i	1.4676 + 0.8746i	2.1581 + 0.6301i	-0.8469 - 0.6548i	-0.0633 + 0.79...
1000	0.6287 + 0.3635i	1.4381 + 0.0362i	1.7187 - 0.3926i	-1.3747 + 0.6045i	-0.5371 - 0.2786i	0.6159 + 0.1963i

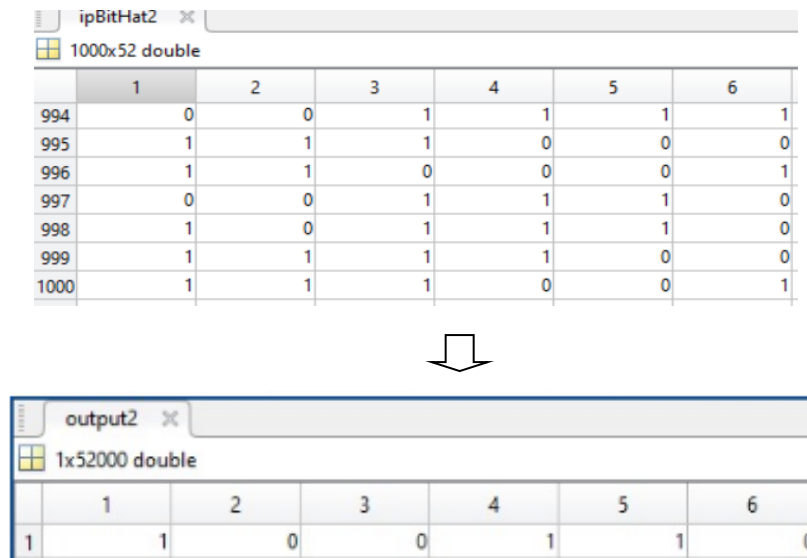


ipBitHat2						
1000x52 double						
	1	2	3	4	5	6
994	0	0	1	1	1	1
995	1	1	1	0	0	0
996	1	1	0	0	0	1
997	0	0	1	1	1	0
998	1	0	1	1	1	0
999	1	1	1	1	0	0
1000	1	1	1	0	0	1

Gambar 3. 19 Output Demodulasi

3.4.13 Parallel to Serial (P/S)

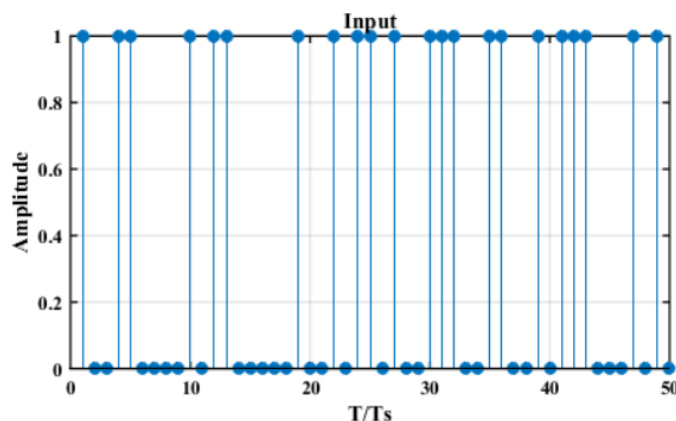
Berdasarkan Gambar 3.2, bit informasi dari blok *demapper* (*yd*) yang masih berupa matriks 1000x52 diubah kembali ke bentuk semula dengan cara konversi dari bentuk *parallel* ke bentuk serial. Pada Matlab, konversi dari bentuk *parallel* ke bentuk serial menggunakan fungsi *reshape* sehingga menghasilkan keluaran berupa matriks 1x52000 seperti pada Gambar 3.20.



Gambar 3. 20 Output Parallel to serial pada Receiver

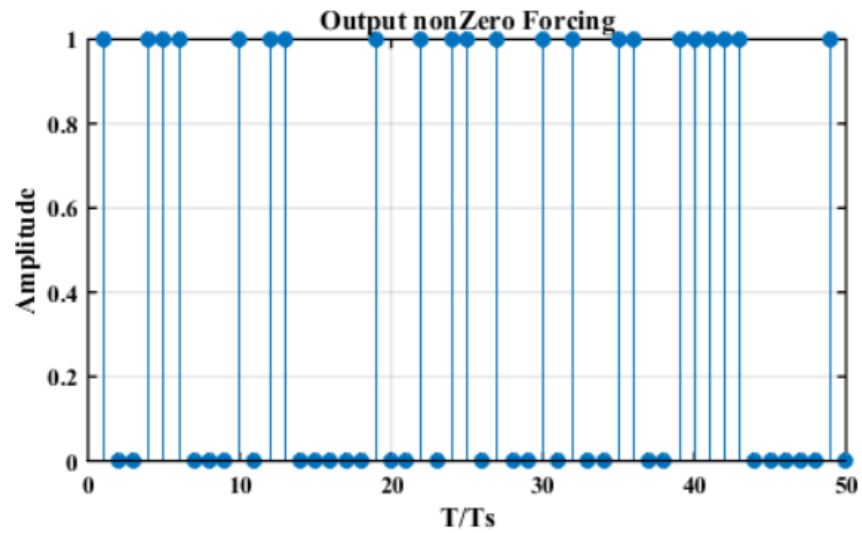
3.4.14 Data keluaran

Output bit menggunakan ekuliasasi *zero forcing* dan *non zero forcing* menjadi 1x52000. Hasil keluaran bit akan digunakan untuk mengetahui nilai kesalahan bit yang diterima di sisi *receiver* yang dibandingkan dengan nilai *Bit Error Rate* (BER).



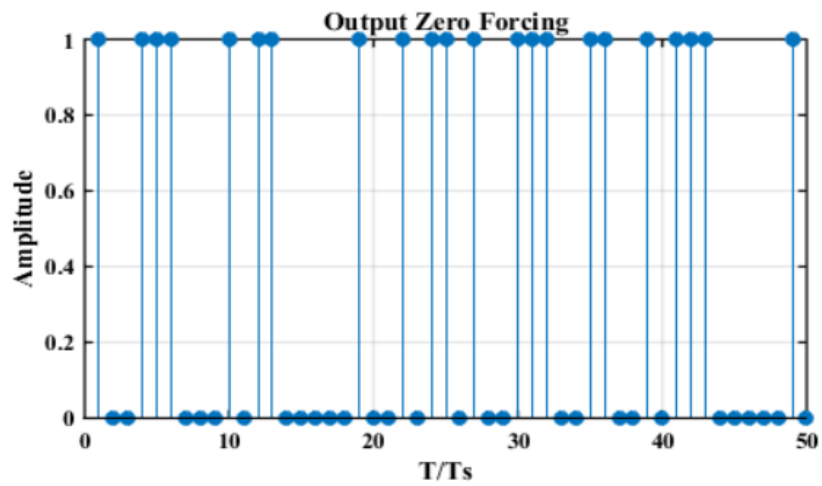
30	31	32	33	34	35	36	37	38	39	40
1	1	1	0	0	1	1	0	0	1	0

Gambar 3. 21 *Output Bit Input*



31	32	33	34	35	36	37	38	39	40
0	1	0	0	1	1	0	0	1	1

Gambar 3. 22 *Output NonZero Forcing*



30	31	32	33	34	35	36	37	38	39	40
1	1	1	0	0	1	1	0	0	1	0

Gambar 3. 23 *Output Zero Forcing*

3.4.15 Simulasi *Bit Error Rate* (BER)

Pada tahap ini dilakukan perhitungan BER pada sistem OFDM menggunakan kanal *Correlated Double Ring* dengan ekualisasi *Zero Forcing* dan *Non Zero Forcing* untuk mengetahui kinerja dari sistem yang telah dirancang. Perhitungan BER simulasi merujuk pada persamaan 2.9 dan perhitungan BER teoritis merujuk pada persamaan 2.41.