

BAB III

METODE PENELITIAN

3.1 ALAT YANG DIGUNAKAN

Pengkajian ini menggunakan kanal komunikasi antar kendaraan yaitu kanal V2V dengan *scatterer* yang bergerak. Terdapat tiga tingkatan kecepatan untuk *scatterer*, yaitu kecepatan lambat, sedang, dan cepat dengan jumlah 8 *scatterer* yang terlibat. Sistem *multi carrier* diimplementasikan sebagai OFDM. Memitigasi *Doppler effect*, menggunakan teknik *Zero Forcing Equalization* diterapkan, serta dampak dari gabungan yang diverifikasi melalui parameter BER. Simulasi dilakukan menggunakan perangkat lunak MATLAB R2018a.

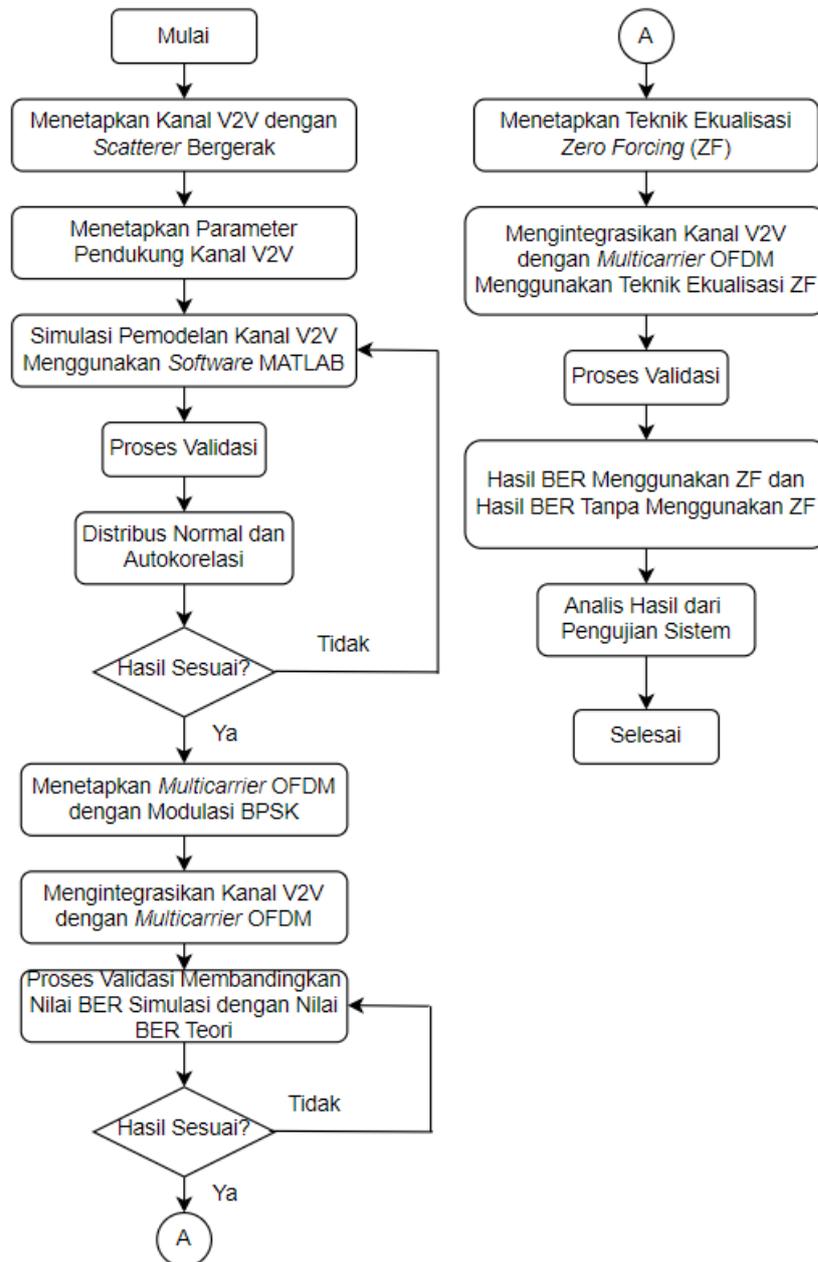
3.2 ALUR PENELITIAN

Gambar 3.1 adalah alur dari pengkajian yang menggambarkan langkah-langkah yang diambil dalam penelitian ini. Proses penelitian ini terbagi menjadi beberapa tahap. Tahap pertama penentuan model kanal, dengan fokus pada kanal V2V *scatterer* bergerak. Dalam tahap ini, parameter penunjang ditambahkan total *scatterer* bergerak sebanyak 8, ragam bentuk kecepatan 7 m/s, 17 m/s, dan 27 m/s untuk pengirim dan penerima, dan 5 m/s untuk kecepatan *scatterer*. Ditambahkan *noise* AWGN. Tahap kedua mencakup pemodelan kanal memakai perangkat lunak MATLAB.

Proses selanjutnya, yakni tahap ketiga, melibatkan tahapan validasi memakai distribusi normal dan autokorelasi. Jika hasil validasinya tidak akurat, dengan demikian penelitian kembali ke proses simulasi pemodelan kanal V2V dengan *scatterer* bergerak, kemudian tahap validasi hasil simulasi. Jika hasil validasi sesuai teori, langkah selanjutnya adalah tahap keempat, di mana sistem *multi carrier* yang ditetapkan adalah OFDM dengan modulasi BPSK. Tahap kelima yaitu integrasi model kanal V2V dengan *scatterer* bergerak ke dalam sistem *multi carrier* OFDM.

Proses ke enam penentuan teknik ekualisasi, di mana ekualisasi *Zero Forcing* dipilih. Setelah melakukan integrasi kanal V2V pada *multi carrier* OFDM, teknik *equalization* diterapkan guna memitigasi *Doppler Effect*. Tahap ke tujuh melakukan pengujian hasil integrasi menggunakan Matlab, dan jika hasilnya tidak sesuai

dengan teori, dilakukan perbaikan program. Proses akhir adalah analisis pengaruh kecepatan *scatterer* bergerak atau *Doppler effect* terhadap parameter BER yang diperoleh oleh kanal V2V memakai *multi carrier* OFDM dengan teknik *Zero Forcing Equalization*. *Flowchart* bisa dilihat pada Gambar 3.1.



Gambar 3.1 Flowchart Alur Penelitian

3.3 PARAMETER SIMULASI

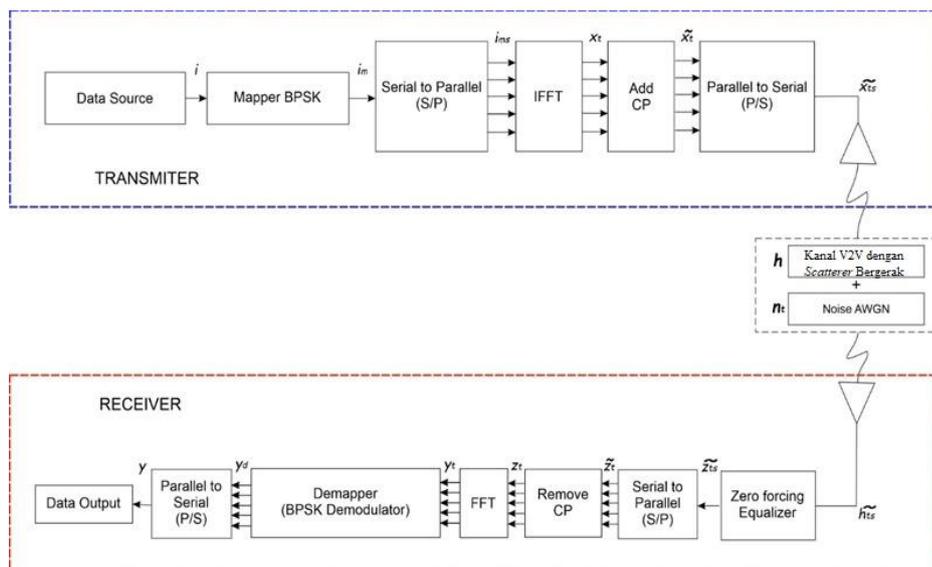
Tabel 3.1 dibawah ini adalah parameter yang digunakan pada simulasi.

Tabel 3. 1 Parameter Simulasi

Parameter	Simbol	Nilai
Jenis Modulasi		BPSK
Jumlah Level Modulasi	ml	1
Bit Input	i	52000 bit
Jumlah FFT	Tb	64
Jumlah Simbol	nSym	1000
Kanal	h	V2V
Ekualisasi		Zero Forcing
Frekuensi Pembawa	f_c	$5,8 \times 10^9$ Hz
Frekuensi Sampling	f_s	40000 Hz
Kecepatan Kendaraan	$V_t = V_r$	7 m/s, 17 m/s dan 27 m/s
Laju Cahaya	c	3×10^8
Kecepatan Scatterer	V_s	5 m/s
Jumlah Scatterer	n	8
Sudut antara garis horizontal dengan arah gerak kendaraan pengirim (Tx)	α_{vt}	5°
Sudut antara garis horizontal dengan arah gerak kendaraan penerima (Rx)	α_{vr}	5°
Fasa	θ	5°

3.4 PEMODELAN SISTEM

Gambar 3.2 menunjukkan pemodelan sistem utama OFDM dengan modulasi BPSK.

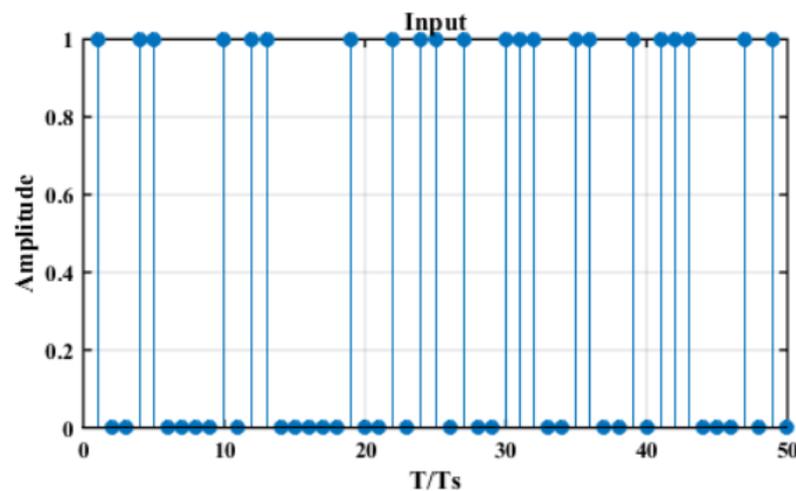


Gambar 3. 2 Pemodelan Sistem OFDM

3.4.1 Data Masukan

Data masukan di *block* diagram simulasi sistem berasal dari *data source*. Fungsi utama dari *data source* ini adalah untuk membangkitkan data biner yang akan dijadikan sebagai informasi yang dikirim. Dalam kajian ini, jumlah bit informasi (*i*) dihasilkan sejumlah 52000 bit. Nilai dari data yang dihasilkan disesuaikan dengan level modulasi yang digunakan.

Proses pembangkitan bit informasi dilakukan secara acak dengan memakai fungsi *randsrc* di perangkat lunak Matlab R2018a. Hal ini menghasilkan pola acak dari bit 1 dan bit 0, dengan kemungkinan muncul yang sama untuk keduanya. Sinyal bit informasi yang dibangkitkan dilihat pada Gambar 3.3.



Gambar 3. 3 Sinyal Masukan

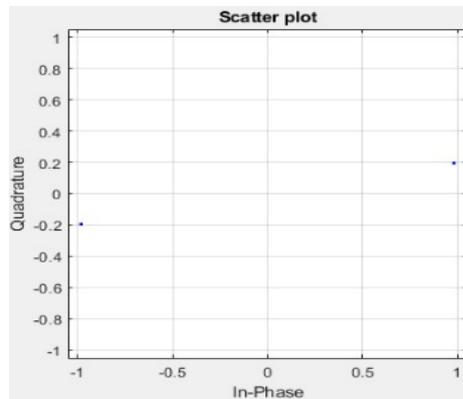
ipBit						
<input checked="" type="checkbox"/> 1x52000 logical						
	1	2	3	4	5	6
1	0	0	0	1	1	0

Gambar 3. 4 Bit Biner Sinyal Masukan

3.4.2 Mapper BPSK

Tahap ini, bit informasi (*i*) dibangkitkan akan mengalami proses pengiriman yang dikodekan ke dalam BPSK *modulation*. Dalam BPSK, *phase* dari *frequency carrier* diperbaharui antara dua nilai yang merepresentasikan kondisi biner 1 dan 0. Pada konteks ini, *phase* dari *frequency carrier* satu dengan yang lain memiliki

perbedaan sebesar π radian (180°). Diagram konstelasi modulasi BPSK ditunjukkan Gambar 3.5.



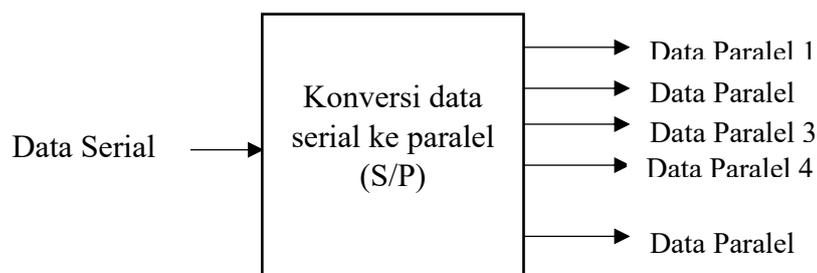
Gambar 3. 5 Diagram Konstelasi BPSK

ipMod						
1x52000 double						
	1	2	3	4	5	6
1	-1	-1	-1	1	1	-1

Gambar 3. 6 Mapper BPSK

3.4.3 Serial to Parallel (S/P)

Block serial ke paralel berperan dalam mengganti aliran data dari 1 baris menjadi sejumlah baris dan kolom. Tahapan serial ke paralel dijalankan setelah bit informasi dimodulasi menjadi simbol data. Simbol data (*im*) sejumlah 1x52000 dikirimkan dengan cara serial dan selanjutnya diubah dalam bentuk paralel sesuai dengan total *subcarrier* yang dipakai. Total *subcarrier* yang dipakai dalam penelitian ini adalah 52, maka setiap *subcarrier* memiliki 1000 simbol data. Keluaran dari (*ims*) berbentuk matriks 1000x52. Dalam pemodelan ini, tahapan dari serial ke paralel dijalankan melalui fungsi *reshape* di perangkat lunak MATLAB.



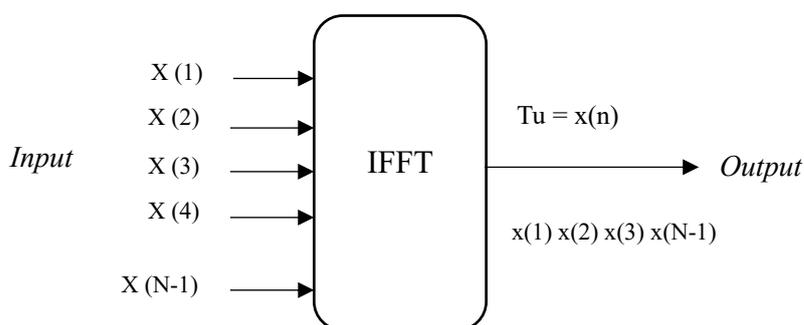
Gambar 3. 7 Pengubah Data Serial ke Paralel

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

Gambar 3. 8 Output Serial to Parallel sisi TX

3.4.4 Proses Invers Fast Fourier Transform (IFFT)

IFFT adalah tahapan supaya mendapatkan *symbol* OFDM di bagian pengirim. Dalam proses ini, frekuensi dari masing-masing informasi dibentuk saling *orthogonal*, sehingga *sub*-kanal bisa saling *overlapping* tanpa menyebabkan interferensi. Tahap IFFT ini disebut sebagai proses *modulation* pada OFDM. IFFT merubah *spectrum*, yakni *amplitude* dan *phase*, di masing-masing sinyal informasi ke bentuk sinyal pada domain waktu. Data yang digunakan pada tahap IFFT berbentuk paralel, disajikan dalam bentuk matriks baris serta kolom. Besaran N dipakai untuk IFFT sebanyak 64, di mana komponen baris merepresentasikan data dari *subcarrier*, dan kolom merepresentasikan data dari simbol yang dibangkitkan. Sebagai hasil *output* dari IFFT ($x(t)$) adalah matriks berukuran 1000x64. Dalam pemodelan ini, proses IFFT dijalankan memakai fungsi "ifft" pada MATLAB.



Gambar 3. 9 Bagan IFFT

xF																					
1000x64 double																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	0	0	0	0	0	0	0	-1	-1	-1	1	1	-1	1	-1	-1	-1	1	1	-1	-1
2	0	0	0	0	0	0	0	1	-1	1	1	-1	1	-1	-1	1	1	1	-1	1	-1
3	0	0	0	0	0	0	0	1	1	1	-1	1	-1	1	-1	-1	-1	-1	-1	-1	-1
4	0	0	0	0	0	0	0	1	-1	1	1	-1	-1	-1	1	1	1	1	1	1	-1
5	0	0	0	0	0	0	0	-1	-1	1	-1	1	1	-1	1	-1	-1	1	1	1	-1
6	0	0	0	0	0	0	0	1	-1	-1	-1	1	1	1	-1	-1	1	-1	-1	1	-1
7	0	0	0	0	0	0	0	-1	-1	1	1	-1	1	-1	1	1	1	1	1	-1	-1
8	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	1	-1	1	1	1	1	-1	-1
9	0	0	0	0	0	0	0	1	-1	1	-1	-1	-1	-1	-1	1	-1	1	-1	-1	1
10	0	0	0	0	0	0	0	1	1	1	-1	-1	-1	1	1	-1	1	1	1	1	1
11	0	0	0	0	0	0	0	1	1	-1	-1	-1	-1	-1	-1	1	-1	1	-1	1	-1



xt						
1000x64 complex double						
	1	2	3	4	5	6
994	-1.4434 + ...	-0.6166 + ...	0.0272 - 0....	-0.6051 - 0...	0.0052 - 0....	1.3236 - 0....
995	-0.2887 + ...	0.1956 - 0....	0.6826 - 0....	0.0682 - 0....	0.0937 + 0....	-0.0022 + ...
996	1.4434 + 0....	0.3669 - 0....	0.0813 + 0....	-0.4789 + ...	-1.4531 - 1...	-0.1955 + ...
997	-1.1547 + ...	0.1648 - 0....	0.9429 + 0....	-0.1702 - 0...	0.1303 + 0....	0.0009 + 1....
998	0.8660 + 0....	0.8971 + 0....	-0.8222 - 0...	-1.5629 - 0...	-0.3970 + ...	-0.2702 - 0...
999	0.8660 + 0....	-0.2598 - 1...	0.5737 + 0....	1.4150 - 0....	0.0479 - 0....	-0.4681 + ...
1000	0.8660 + 0....	0.4690 - 0....	-0.0471 + ...	0.4884 + 0....	0.8700 - 1....	0.5344 - 0....

Gambar 3. 10 Hasil Keluaran IFFT

3.4.5 Penambahan *Cyclic Prefix* (CP)

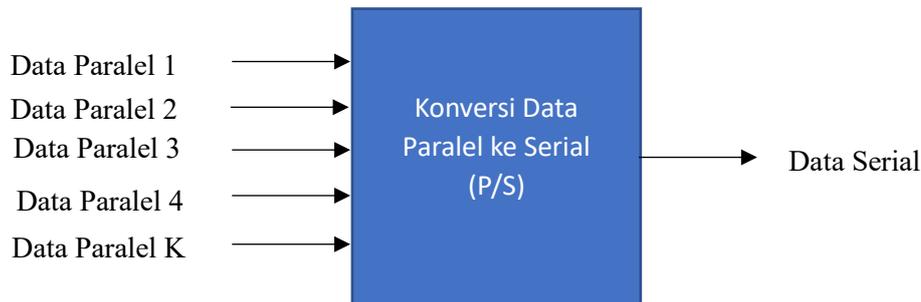
Dalam simulasi ini, keluaran dari IFFT akan ditambah beberapa *guard interval* sebelum ditransmisikan. Proses penambahan CP bertujuan untuk menghindari terjadinya *Intersymbol Interference* (ISI). Pada sistem komunikasi, *guard interval* panjangnya 1/8 atau 1/4 dari total keluaran IFFT. *Guard interval* ditempatkan sebelum simbol yang paling depan dan diisi oleh potongan simbol yang terakhir. Jumlah CP pada simulasi ini adalah 1/4 dari total simbol OFDM. Sebagai contoh, apabila jumlah simbol OFDM adalah 128, maka total CP yang dipakai adalah 32 simbol.

	1	2	3	4	5	6
993	-1.4434 + ...	-0.2905 - 0...	0.3927 - 0....	1.3746 - 0....	1.1199 - 0....	-0.4986 - 0...
994	0.0000 + 0....	-0.6641 - 0...	0.0870 + 0....	0.5420 + 0....	0.5075 + 0....	0.2567 - 0....
995	-0.5774 + ...	-0.3400 - 1...	-0.5048 - 0....	1.0664 + 0....	0.0626 - 0....	-0.9040 - 0...
996	0.5774 + 0....	0.0909 + 0....	-1.2470 + ...	-0.8608 + ...	0.1446 + 0....	0.3948 - 1....
997	0.2887 - 0....	-0.9557 - 1...	-0.6554 - 0....	0.6771 + 0....	1.1367 + 1....	0.0125 + 0....
998	-0.8660 - 0...	-0.9097 - 0....	1.1994 + 0....	-0.7399 + ...	-1.2471 - 0...	-0.2273 + ...
999	0.8660 + 1....	0.3740 + 0....	-0.3955 - 0....	0.1804 - 0....	-0.5813 + ...	-0.7403 + ...
1000	0.2887 + 0....	-0.1899 + ...	-0.6522 - 1...	0.0292 - 0....	-0.0717 + ...	0.6463 + 0....

Gambar 3. 11 Output Penambahan Cyclic Prefix

3.4.6 Parallel to Serial (P/S)

Dalam blok diagram, sinyal OFDM yang telah dikirim kemudian ditambahkan CP perlu diubah dalam bentuk serial. Sinyal OFDM awalnya terdiri dari sejumlah baris dan kolom, setelah dikonversi ke bentuk serial menjadi satu baris dan sejumlah kolom.



Gambar 3. 12 Pengubah Data Parallel to Serial

	1	2	3	4	5	6
1	-0.2887 - 0...	-0.4332 + ...	1.8529 - 0....	1.0328 - 1....	0.7065 - 1....	0.1050 - 0....

Gambar 3. 13 Output dari Parallel to Serial sisi TX

3.4.7 Pemodelan Kanal Vehicle to Vehicle (V2V)

Model kanal pada simulasi ini menggunakan kanal V2V dengan *scatterer* bergerak. Kanal V2V memiliki dua sisi, pengirim dan penerima yang bergerak pada kecepatan stabil dikelilingi *scatterer* yang bergerak. Pengirim, penerima, dan

scatterer bergerak bersama dengan kecepatan *random*. Dalam kanal V2V, arah propagasi dibagi menjadi 2, yakni sinyal dikirim dari penerima ke pengirim tanpa hambatan atau *Line of Sight* (LOS), dan sinyal yang dikirim penerima ke pengirim dipantulkan terlebih dahulu karena adanya *scatterer* bergerak, kemudian sinyal tersebut ditransmisikan ke penerima. Simulasi ini menggunakan kecepatan kendaraan dengan kecepatan rendah 7 m/s, kecepatan sedang 17 m/s, dan kecepatan tinggi 27 m/s. Jumlah *scatterer* yang digunakan adalah 8. Untuk memvalidasi kanal ini, digunakan proses distribusi *Gaussian* atau distribusi normal.

3.4.8 Proses Ekualisasi

Proses ekualisasi adalah langkah untuk menyelidiki respons supaya kanal dapat diatasi. Tahap ini melibatkan pengolahan sinyal digital untuk merekonstruksi kembali sinyal data yang dikirimkan. Dalam pemodelan ini, digunakan proses *equalization* ZF dikarenakan mempunyai kompleksitas paling rendah dibanding dengan metode *equalization* lainnya. Proses *equalization* ZF dijalankan dengan mengalikan *invers* dari kanal (h) dengan *receive* ($\tilde{h}ts$). Keluaran dari ekualisasi *Zero Forcing* pada simulasi sistem ditetapkan dengan ($\tilde{Z}ts$) yang berukuran 1x80000.

	1	2	3	4	5	6
1	-6.5277 + ...	-9.2413 - 1...	2.0088e+0...	-1.1162e+...	-3.4462e+...	-0.7494 + ...

Gambar 3. 14 Output Receive Non ZF

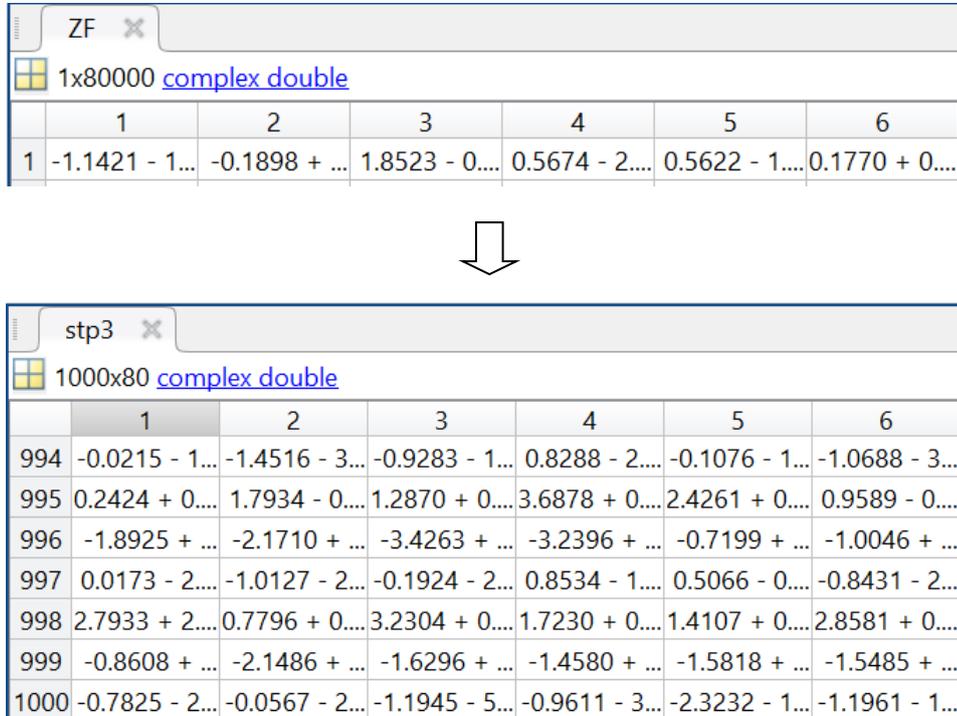
	1	2	3	4	5	6
1	-1.1421 - 1...	-0.1898 + ...	1.8523 - 0....	0.5674 - 2....	0.5622 - 1....	0.1770 + 0....

Gambar 3. 15 Output Receive ZF

3.4.9 Serial to Parallel (S/P)

Di dalam *block* ini, *Output* dari diagram *block* ZF ($\tilde{Z}ts$) yang semula berbentuk serial 1x80000 diubah menjadi bentuk paralel, menghasilkan *output*

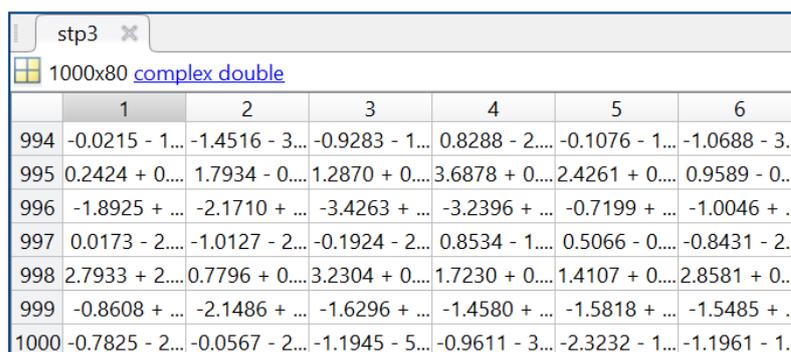
1000x80. Proses konversi dari bentuk serial ke paralel dilakukan melalui penggunaan fungsi *reshape* di Matlab. Keluaran S/P di sisi penerima dari simulasi sistem ditetapkan menggunakan notasi ($\tilde{Z}t$).



Gambar 3. 16 Output S/P ZF pada sisi RX

3.4.10 Menghilangkan *Cyclic Prefix* (CP)

Di bagian penerima, *Cyclic Prefix* akan dihapus untuk mendapatkan data informasi sesungguhnya. Penghilangan CP dijalankan dengan mengeliminasi 1/4 dari total simbol OFDM dari masing-masing *subcarrier*, setara dengan 16 simbol. Jumlah awal sinyal yang sudah ditambahkan CP adalah 80, dan setelah proses pelepasan CP, sinyal akan dikembalikan ke jumlah awal 64, seperti ditunjukkan dalam Gambar 3.17 dalam pemodelan sistem ditetapkan dengan (Zt).





rcp2						
1000x64 complex double						
	1	2	3	4	5	6
994	-4.2779 + ...	-1.5342 + ...	-2.4716 + ...	-2.6105 + ...	-2.3113 + ...	0.6525 + 2...
995	-0.5073 - 2...	-0.0157 - 2...	0.6906 - 1....	-0.1756 - 3...	-1.1915 - 1...	0.1034 - 1....
996	3.3309 + 0....	1.5243 + 0....	1.8711 + 1....	1.8863 + 1....	0.5812 - 2....	2.0300 - 0....
997	-3.1269 + ...	-1.8770 + ...	-0.8554 + ...	-1.3751 + ...	-1.2050 + ...	-1.1179 + ...
998	1.5860 - 2....	1.0516 - 1....	-1.8147 - 1....	-1.9644 - 2....	-1.5442 - 0....	-1.5876 - 3...
999	2.4580 - 0....	2.7359 - 1....	4.2741 + 1....	3.4715 + 0....	2.8528 - 1....	0.8580 + 1....
1000	-1.3663 + ...	-1.8343 + ...	-2.3829 + ...	-1.2723 + ...	-0.7126 + ...	-1.1216 + ...

Gambar 3. 17 Ouput Removing Cyclic Prefix

3.4.11 Proses *Fast Fourier Transform* (FFT)

Proses FFT adalah kebalikannya proses IFFT, dimana FFT bertujuan untuk merubah *time domain* ke *frequency domain*. Tujuan dari tahapan ini adalah membagi antara *frequency* pembawa pada simbol OFDM yang diterima di bagian penerima sebelum di demodulasi serta konversi kembali ke dalam bentuk bit informasi. Pada Matlab, tahapan FFT bisa dijalankan memakai fungsi "fft". Keluaran dari FFT pada Gambar 3.18 dan dijelaskan dengan notasi (yt) berdasarkan Gambar 3.2.

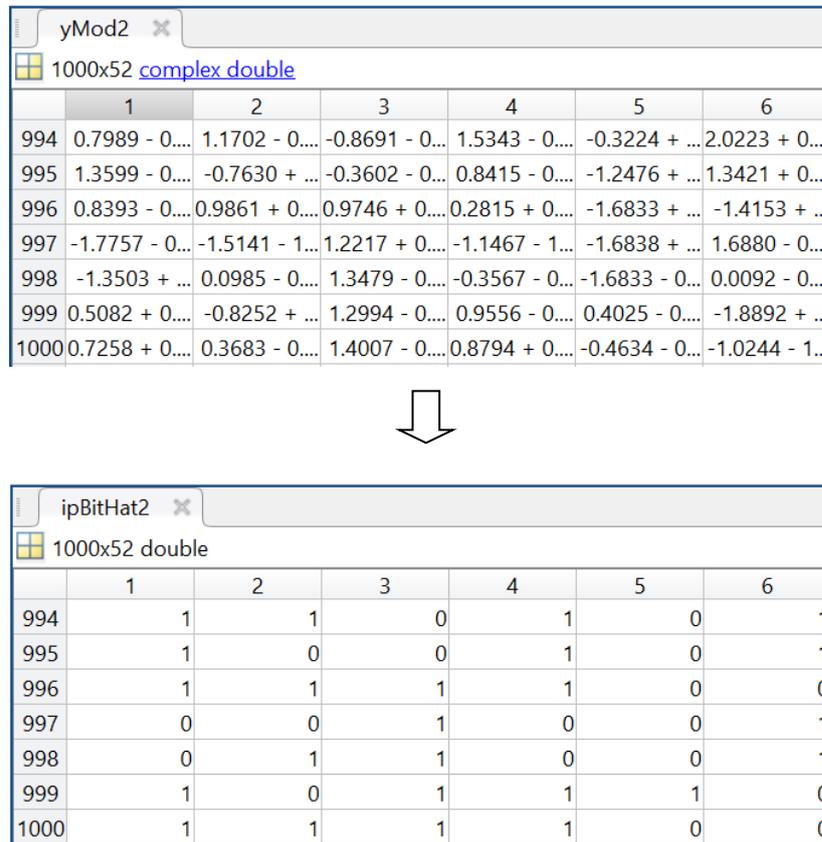
yF2						
1000x64 complex double						
	1	2	3	4	5	6
994	0.1846 + 0....	-0.1831 - 0...	0.3215 + 0....	0.4507 + 0....	-0.0431 - 0...	0.2758 - 0....
995	1.2765 + 0....	-0.4124 + ...	0.0435 - 0....	0.3315 + 0....	0.2933 + 0....	0.3753 - 0....
996	-0.1522 - 0....	0.7297 + 1....	-0.4941 + ...	-0.7552 + ...	0.1181 - 0....	0.3681 + 0....
997	0.6426 + 0....	-0.1282 - 0....	-0.5050 + ...	0.3418 + 0....	0.2068 - 0....	-0.3005 - 0...
998	0.1936 - 0....	-0.0090 - 0....	-0.0865 - 0....	-0.2786 - 0....	0.1207 - 0....	0.3994 - 0....
999	-0.3816 - 0....	0.4117 - 0....	0.2376 + 0....	-0.3248 - 0....	0.2553 + 0....	-1.0765 - 0....
1000	0.4765 - 0....	-0.9404 - 0....	-0.3217 - 0....	-0.4965 + ...	0.3442 + 0....	0.6560 + 0....

yMod2						
1000x52 complex double						
	1	2	3	4	5	6
994	0.7989 - 0....	1.1702 - 0....	-0.8691 - 0...	1.5343 - 0....	-0.3224 + ...	2.0223 + 0....
995	1.3599 - 0....	-0.7630 + ...	-0.3602 - 0....	0.8415 - 0....	-1.2476 + ...	1.3421 + 0....
996	0.8393 - 0....	0.9861 + 0....	0.9746 + 0....	0.2815 + 0....	-1.6833 + ...	-1.4153 + ...
997	-1.7757 - 0....	-1.5141 - 1....	1.2217 + 0....	-1.1467 - 1....	-1.6838 + ...	1.6880 - 0....
998	-1.3503 + ...	0.0985 - 0....	1.3479 - 0....	-0.3567 - 0....	-1.6833 - 0....	0.0092 - 0....
999	0.5082 + 0....	-0.8252 + ...	1.2994 - 0....	0.9556 - 0....	0.4025 - 0....	-1.8892 + ...
1000	0.7258 + 0....	0.3683 - 0....	1.4007 - 0....	0.8794 + 0....	-0.4634 - 0....	-1.0244 - 1....

Gambar 3. 18 Hasil Keluaran FFT

3.4.12 Demodulasi

Dalam tahap demodulasi, sinyal keluaran dari FFT (y_t) dipisah antara sinyal *carrier* dan sinyal data. Selanjutnya, gelombang data terdeteksi akan dipulihkan ke bentuk deret data biner, sebagaimana tampak di Gambar 3.19.



Gambar 3. 19 Hasil Keluaran Demodulasi

3.4.13 Parallel to Serial (P/S)

Mengacu pada Gambar 3.2, bit informasi dari *block demapper* (y_d) masih dalam bentuk matriks 1000x52 dikembalikan ke bentuk awal dengan mengkonversinya dari paralel ke serial. Di Matlab, konversi dari bentuk paralel ke serial dapat dijalankan dengan fungsi *reshape*, diperoleh *output* berupa matriks 1x52000, sebagaimana terlihat di Gambar 3.20.

