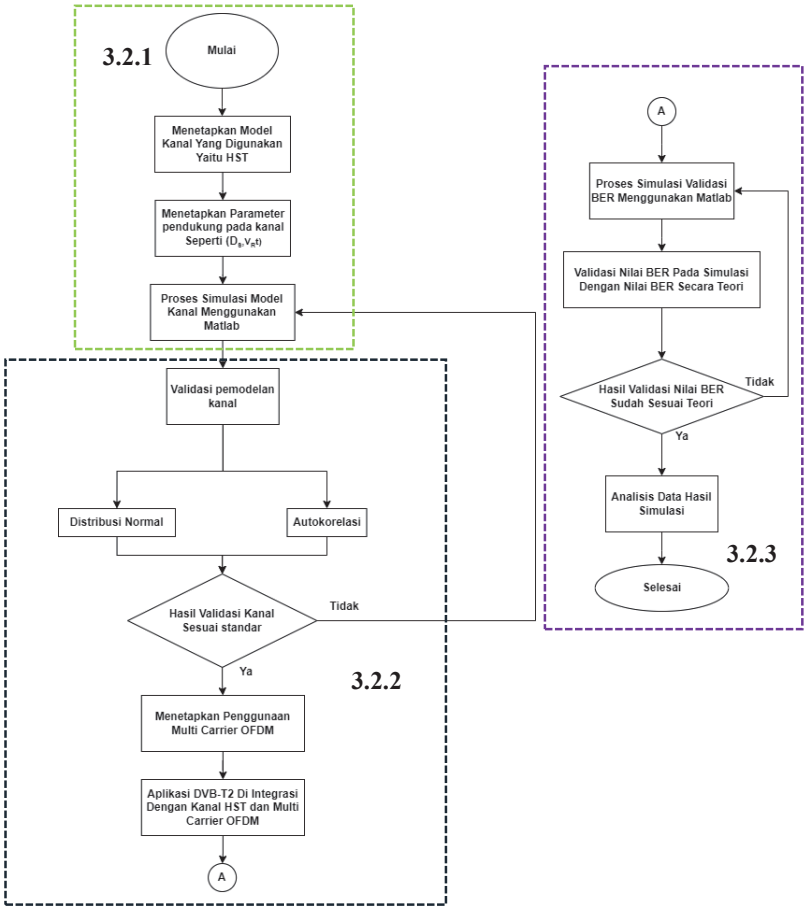
BAB III
METODE PENELITIAN

3.1 ALAT DAN BAHAN

Penelitian ini menggunakan suatu pemodelan program simulasi untuk menganalisis aplikasi dari DVB-T2 pada sistem komunikasi *High Speed Train.* Simulasi dilakukan dengan *multicarrier* dengan level 64-QAM pada kanal HST agar dapat melihat nilai BER akibat efek *multipath* maupun efek *Doppler*. Model simulasi yang diimplemetasikan dalam penelitian ini menggunakan *simulink* pada MATLAB R2022b.

3.2 ALUR PENELITIAN

Pada alur penelitian membahas beberapa sub pokok bahasan diantaranya *flowchart* penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* Alur Penelitian

Blok diagram dari sistem OFDM dengan level modulasi 64-QAM, parameter-parameter yang digunakan untuk simulasi sistem, sekenario simulasi dan cara memvalidasi hasil simulasi yang akan dibandingkan dengan teori. Kanal yang digunakan yaitu *High Speed Train* (HST). Kecepatan mobilitas pada HST yaitu 300 km/jam. Parameter unjuk kerja pada penelitian ini didasarkan pada nilai BER yang didapatkan. Untuk simulasi sistem ini mengasumsikan kondisi kanal menggunakan *Multicarrier* OFDM, dimana penerima mengetahui informasi bit yang akan dikirimkan. Sedangkan untuk alokasi frekuensi *carrier* yaitu sebesar 700 MHz.

3.2.1 Menetapkan Parameter Yang Digunakan

Pada Gambar 3.1 yang merupakan *flowchart* alur penelitian yang akan dilakukan. Hal pertama yang dilakukan yaitu *study literature* yang telah dipaparkan pada kajian pustaka dan dasar teori. Kemudian masuk pada tahap perancangan, di mana pada tahap ini yaitu menetapkan model kanal yang akan digunakan. Untuk model kanal yang digunakan yaitu *High Speed Train* (HST). Setelah menetapkan model kanal yang digunakan, kemudian akan menentukan parameter pendukung untuk dapat melakukan simulasi. Selanjutnya melakukan simulasi kanal HST menggunankan *simulink* pada Matlab R2022b.

3.2.2 Validasi Kanal Dan Menetapkan *Multicarrier* OFDM

Pada tahap validasi dilakukan dengan dua tahapan yaitu distribusi dan autokorelasi. Pada tahap distribusi dilakukan dengan metode distribusi normal sedangkan pada autokorelasi merupakan proses validasi dengan membandingkan dua sinyal yang sama pada waktu yang berbeda. Jika hasil validasi sudah sesuai dengan teori, maka tahap selanjutnya menetapkan *multicarrier* yang digunakan yaitu *multicarrier* OFDM. Kemudian pada tahapan selanjutnya melakukan integrasi pada aplikasi DVB-T2 dengan kanal HST menggunakan *multicarrier* yang sudah ditetapkan yaitu OFDM.

3.3.3 Proses Simulasi BER dan SNR

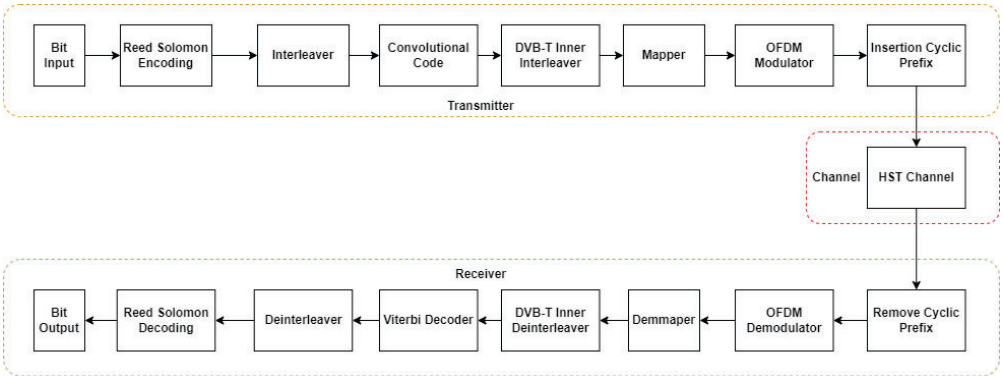
Pada tahapan selanjutnya, melakukan validasi nilai BER dari hasil integrasi pada tahapan sebelumnya. Jika nilai BER yang didapatkan sudah sesuai dengan

teori maka akan dilanjutkan ke tahapan selanjutnya dan jika belum sesuai maka akan dilakukan simulasi kembali. Hal terakhir akan dilakukan analisis data dari hasil yang didapatkan. Hasil yang didapatkan akan berupa grafik.

Untuk parameter pada *multicarrier* OFDM akan ditentukan berdasarkan spesifikasi pada DVB-T2. Modulasi yang digunakan yaitu QAM dengan level modulasi 64-QAM serta *cyclic prefix* yang digunakan sebesar $\frac{1}{4}$ dari nilai FFT *size* sebesar 512. Untuk nilai *Fast Forier Transform* (FFT) yaitu 2K (K=1024), sehingga nilai FFT yang digunakan sebesar 2048. *Channel bandwidth* yang digunakan yaitu 7 MHz.

3.3 BLOK DIAGRAM SISTEM DVB – T2

DVB-T2 merupakan generasi terbaru dari generasi sebelumnya yaitu DVB-T yang memiliki banyak keunggulan. Diantaranya memiliki kualitas gambar yang lebih tajam modulasi yang beragam serta penggunaan *multicarrier* OFDM merupakan alasan evolusi dari DVB-T. Gambar 3.2 merupakan blok diagram untuk penelitian pada DVB-T2 yang sudah diintegrasikan pada *multicarrier* OFDM dengan level modulasi 64-QAM.



Gambar 3.2 Rancangan Blok Diagram Sistem Komunikasi DVB – T2 Pada kanal HST

Pada bagian pengirim atau *transmitter* memiliki beberapa blok yaitu *bit stream* dimana bit-bit dibangkitkan yang memiliki nilai 0 dan 1 bit, terdapat blok QAM *Mapper* yang digunakan untuk melakukan proses penumpangan pada frekuensi pembawa sehingga meminimalisasikan kerusakan pada informasi yang

dikirimkan. Pada bagian level QAM akan menggunakan modulasi 64-QAM. Setelah melakukan proses modulasi, dilakukan konversi *serial to parallel* untuk mengubah informasi bit secara serial menjadi informasi paralel. Simbol-simbol ini disisipi pilot pada semua *subcarrier* sepanjang satu simbol sesuai dengan aturan penyusunan pilot tipe blok (*block type*). *Cyclic prefix* yang digunakan adalah 1/4 dari total FFT *size* yaitu 512. Selanjutnya terdapat blok *Inverse Fast Forier Transform* (IFFT) di mana mengubah bentuk data dari domain frekuensi ke domain waktu. Kemudian dilakukan proses *cylic prefix* proses ini dilakukan untuk meminimalisasikan *Intersymbol Interference* (ISI) dan diubah dalam bentuk serial. Kemudian bit – bit yang sudah diproses akan melewati kanal HST.

Setelah melewati bagian kanal terdapat blok yang akan mengubah ke bentuk *parallel* serta memisahkan CP. Kemudian akan dilakukan proses FFT pada sisi penerima serta melakukan. Pada bagian penerima atau *receiver* memiliki proses yang berkebalikan dari blok pengirim. Setelah melewati kanal, akan dilakukan pengecekan pada blok QAM *Detector.* Kemudian menghapus rasio pilot yang sudah dimasukkan serta mengubahnya dalam bentuk *parallel*. Setelah itu dilakukan proses demodulasi pada demodulator untuk dapat memisahkan antara sinyal pembawa dengan sinyal informasi pada sisi *demapper* QAM sehingga dapat menampilkan data nilai bit yang diterima.

3.4 PARAMETER SIMULASI

Adapun parameter simulasi yang akan digunakan pada simulasi dan analisis data untuk kondisi yang akan diuji terdapat pada Tabel 3.1.

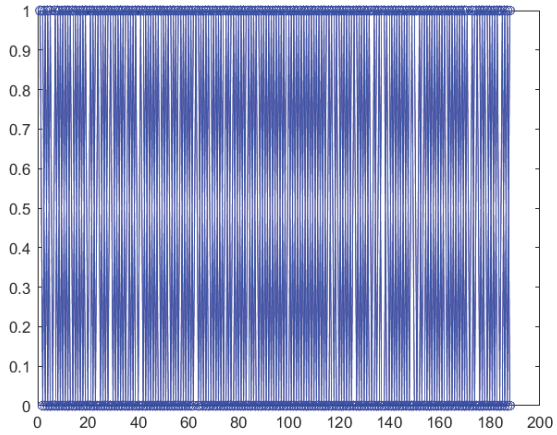
Tabel 3.1 Parameter Simulasi

No.	Parameter	Simbol	Nilai
1	Tipe Modulasi	M	64 – QAM
2	Bit Input	*b*	1504
3	Jumlah *Subcarrier*	L	2048
4	Frekuensi Pembawa	f_c	700 x 10^6 Hz
5	FFT *Size*	N	2048
6	Kanal	h	HST
7	Kecepatan HST	V	300 km/jam
8	*Cyclic Prefix*	CP	¼ x 2048 = 512
9	Jumlah penghambur pada sisi pengirim dan penerima	Nii	4 dan 11

No.	Parameter	Simbol	Nilai
10	Perbandingan daya yang diterima dan daya yang dipantulkan	K	6
11	Sudut kedatangan dan sudut pengiriman	γ_R	Bernilai bilangan bulat acak diantara (-180) – 180 dengan distribusi seragam
12	Sudut antara arah pergerakan kendaraan pengirim dengan garis horisontal	β_T	60^0
13	Sudut antara arah kendaraan penerima dengan garis horisontal	β_R	60^0
14	Frekuensi *sampling*	f_s	2000 Hz

3.5 Data Masukan

Data masukan pada blok diagram rancangan sistem ini merupakan suatu pembakit data (data generator). Data generator ini memiliki fungsi untuk membangkitkan data berbentuk desimal sebagai data masukan yang akan dikirimkan. Pada penelitian ini bit informasi yang dibangkitkan berjumlah 1504 bit. Data yang dibangkitkan sesuai dengan level modulasi yang digunakan. Bit tersebut dibangkitkan menggunakan blok *random bit* pada *simulink* Matlab 2022b, sehingga menghasilkan bit acak 1 dan 0. Sinyal yang dibangkitkan dapat dilihat pada Gambar 3.2 dan Tabel 3.3.



Gambar 3.3 Masukan dalam bentuk bit

Tabel 3.2 Bit masukan

No	Bit Masukan 1x1504												
1	0	1	1	0	1	0	0	0	1	0	1	1	0

3.6 *Reed Solomon Encoding*

Blok *Reed Solomon encoding* merupakan salah satu *channel coding* yang digunkan pada aplikasi DVB-T2. Pada penelitian ini jumlah simbol yang digunakan yaitu 204 simbol sebagai *codeword* (n) dan 188 simbol sebagai bit yang akan dikodekan (k). Simbol tambahan (*redudant*) sebanyak 8 (t) yang merupakan simbol *error* yang mampu diperbaiki. Dimana setiap simbol memiliki 8 bit/simbol (m). Tabel 3.3 merupakan hasil keluaran dari proses *Reed Solomon.* Pada *simulink* menggunakan blok "RS *Encoder*".

Tabel 3.3 Keluaran *Reed Solomon*

No	Reed Solomon 204x1
1	56
2	183
3	43
4	128
…	…
200	95
201	48
202	70
203	152
204	166

3.7 *Interleaver*

Interleaver adalah sebuah proses yang yang dilakukan untuk mengacak suatu bit agar menghindari adanya *burst error.* Pada penelitian ini sesuai dengan panjang simbol sebesar 204 maka elemen yang digunakan untuk melakukan *interleaver* sebanyak simbol juga. Oleh karena itu masing – masing simbol memiliki 8 bit maka nilai 8 bit dikalikan dengan 204 mendapatkan hasil 1632. Sesuai dengan Tabel 3.4 keluaran dari bentuk desimal menjadi bentuk biner.

Tabel 3.4 *Output* setelah *interleaver*

No	Deinterleaver 1632x1
1	0
2	0
3	1
…	…
1629	0
1630	0
1631	0
1632	0

3.8 *Convolutional Code*

Setelah melakukan *interleaver* (pengacakan bit), langkah selanjutnya yaitu *Convolutional code* merupakan pengkodean data yang dilakukan secara konvolusi menggunakan sebuah *shift register* (register geser) dan logika *XOR* yang menampilkan *Mod-2.* Pada blok ini akan dilakukan juga *puncture* dimana teknik ini digunakan dalam komunikasi digital dan pengkodean saluran untuk mengurangi kecepatan kode (*code rate*) dengan menghilangkan beberapa bit tertentu dari aliran data yang dikirimkan. Tabel 3.5 merupakan keluaran dari *convolutional code* dimana bit keluaran dari *interleaver* akan di *puncured* dengan bit [1 1 0 1 1 0]' dan menghasilkan keluaran 2176x1.

Tabel 3.5 Setelah melakukan *convolutional code*

No	Convolutional Code 2176x1
1	0
2	1
3	1
…	…
2173	0
2174	0
2175	1
2176	1

3.9 DVB – T *Inner Interleaver*

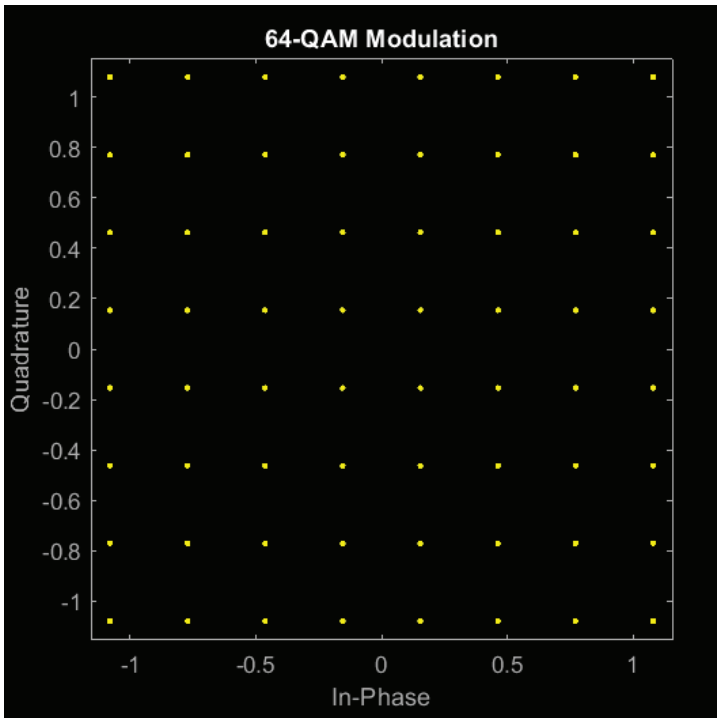
Digital Video Broadcasting – terrestrial (DVB - T) memilki komponen yang disebut *inner interleaver.* Komponen ini merupakan proses pengkodean sinyal dalam DVB – T. Sama seperti proses sebelumnya yaitu *interleaver* akan tetapi pada proses ini memilki perbedaan yaitu mengacak dan menyusun kembali bit – bit data agar meningkatkan ketahanan sinyal pada saat transmisi. Proses *interleaving* melibatkan pengelompokan bit – bit data ke dalam blok – blok yang lebih besar. Hal ini dapat mengurangi *burst error* pada saat pengiriman sinyal. Oleh karena itu bit yang dikirimkan akan meningkat atau lebih banyak dibandingkan bit yang sebelumnya dapat dilihat pada Tabel 3.6.

Tabel 3.6 DVB – T *inner interleaver*

No	DVB-T Interleaver 9072x1
1	0
2	1
3	1
…	…
9069	0
9070	1
9071	1
9072	1

3.10 *Symbol Mapping* 64-QAM

Diagram konstelasi 64-QAM adalah proses merubah bentuk dari biner ($\vec{d}$) menjadi bilangan kompleks menggunakan rumus $S_k = I_k + jQ_k$, dimana k adalah variabel simbol, I merupakan *inphase* atau bilangan *real* dan Q merupakan *quadrature* atau bilangan imajiner.



Gambar 3.4 Diagram konstelasi 64-QAM

Modulasi yang dipergunakan dalam simulasi ini adalah 64-QAM dimana pemetaannya terdiri dari 64 bit. Gambar 3.4 merupakan bentuk simbol *mapping*

modulasi 64-QAM dalam bentuk bilangan kompleks. Jarak antar simbol baik sumbu horisontal (x) maupun vertikal (y) yaitu 3,08607 x 10^{-1} dan 3,08603 10^{-1}.

Tabel 3.7 *Symbol mapping* 64-QAM

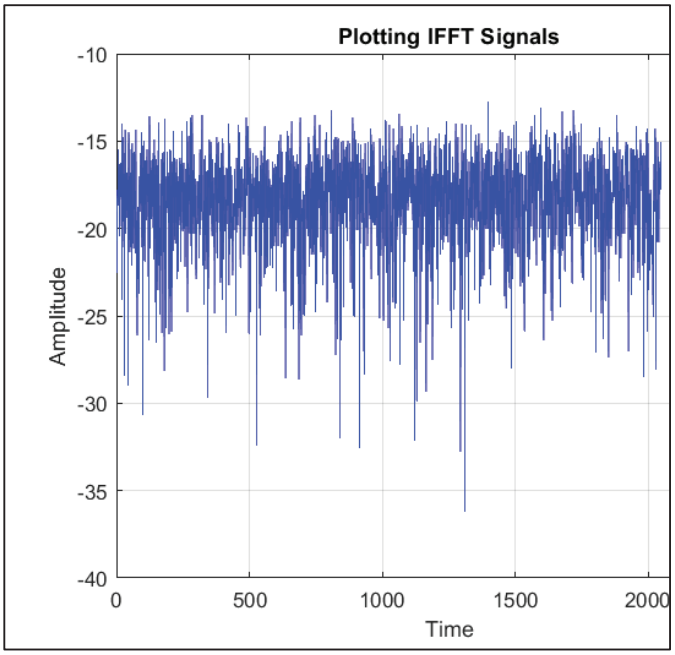
No	Mapper 64-QAM 1512x1
1	-0.4629 + 0,1543i
2	0,1543 – 0,1543i
3	-0,1543 – 0,4629i
…	…
1509	0,4629 – 0,4629i
1510	-0,1546 + 1,0801i
1511	1,0801 + 0,4629i
1512	-0,4629 – 0,7715i

3.11 OFDM Modulator (IFFT)

Setelah melakukan proses modulasi, dilakukan proses *Inverse Fast Fourier Transform* (IFFT) dimana proses ini mengubah bentuk domain frekuensi menjadi dalam domain waktu. Dimana simbol yang dihasilkan dari proses ini saling tegak lurus (*Orthogonal*) dan subkanal dapat saling *overlapping* tanpa menimbulkan *interference*. Data yang dibutuhkan untuk melakukan proses IFFT yaitu *serial,* sehingga dalam bentuk matriks baris seperti yang ditunjukkan pada Tabel 3.5. Pada Gambar 3.11 merupakan sinyal yang dihasilkan sebelum *cyclic prefix.* Pada simulasi ini proses IFFT dilakukan dengan menggunakan blok "OFDM Modulator" pada Matlab *simulink*.

Tabel 3.8 *Output* dari IFFT

No	OFDM Modulator 2048x1
1	0,0030 – 0,0046i
2	-0,0084 – 0,0120i
3	-0,0122 + 0,0024i
…	…
2045	0,0104 – 0,0135i
2046	-0,0057 – 0,0158i
2047	-0,0307 + 0,0045i
2048	-0,0194 + 0,0047i



Gambar 3.5 *Output* dari IFFT

Total *Guard Interval* (GI) yang digunakan pada *size* FFT 2048 yaitu 536 dengan jumlah *carrier* yang membawa informasi yaitu 1512 *carrier.* Amplitudo yang didapatkan berubah setiap waktu hal ini disebabkan perubahan kanal setiap waktu dan pengaruh *multipath* dan efek *Doppler.*

3.12 *Cyclic Prefix Insertion*

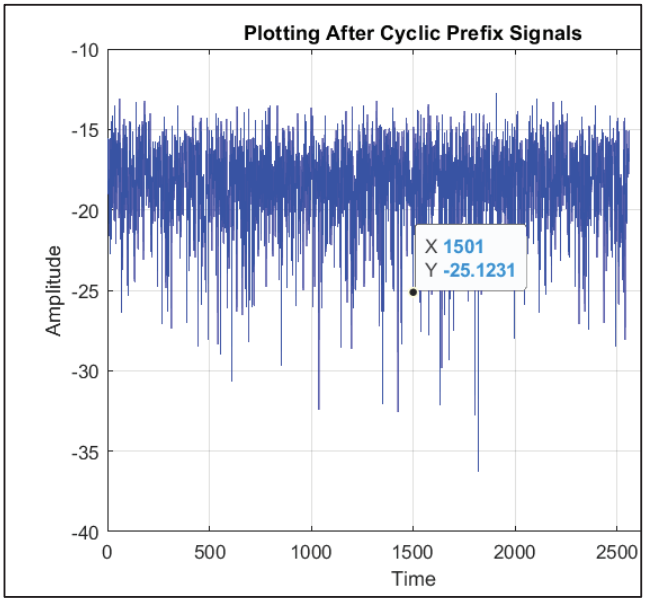
Sistem OFDM, *Cyclic Prefix* (CP) memiliki fungsi yang digunakan untuk mengatasi efek *Intersymbol Interference* (ISI) yang diakibatkan oleh kanal *multipath fading.* Simbol yang telah diberikan CP memiliki kemampuan untuk dipulihkan dengan baik oleh penerima meskipun mengalami gangguan *fading* yang signifikan pada kanal. CP merujuk pada bagian awal simbol OFDM yang berisi replikasi dari bagian akhir simbol OFDM. Jumlah CP yang digunakan adalah sekitar seperempat dari jumlah simbol OFDM. Tabel 3.9 merupakan tambahan simbol yang dihasilkan.

Tabel 3.9 Penambahan *Cyclic Prefix* (CP)

No	Adding Cylic Prefix 2560x1
1	-0,0054 – 0,0113i
2	0,0096 + 0,0084i

No	*Adding Cylic Prefix* 2560x1
3	0,0173 + 0,0216i
…	…
2557	0,0104 – 0,0135i
2558	-0,0057 – 0,0158i
2559	-0,0307 + 0,0045i
2560	-0,0194 + 0,0047i

Gambar 3.6 merupakan keluaran sinyal setelah ditambahkan simbol pada bagian paling depan simbol. Perbedaan denga Gambar 3.5 dapat dilihat dari sumbu x (*time)* yang memiliki nilai yang lebih panjang. Hal ini diakiibatkan karena pencuplikan bagian akhir simbol dan letakkan pada bagian depan simbol.



Gambar 3.6 Sinyal setelah *Cyclic Prefix*

3.13 Kanal HST

Kanal yang digunakan pada penelitian adalah kanal *High Speed Train* (HST) ($\widetilde{H}$) dengan kecepatan tinggi yaitu 300 km/jam. Pada kanal HST mempunyai satu sisi yang bergerak yaitu penerima saja (Rx) sedangkan pengirim (Tx) tidak bergerak. Arah propagasi yang digunakan pada kanal HST dibagi menjadi 2 jenis yaitu dari pengirim menuju ke penerima tanpa adanya *obstacle* (LoS) dan sinyal yang dikirimkan terpantul – pantul (*multipath components*) menuju ke penerima. Jumlah *scatter components* yang digunakan pada penelitian ini sebanyak 50.

Validasi yang dilakukan pada kanal ini menggunakan proses distribusi *gausian* dan distribusi normal. Tabel 3.10 merupakan kanal HST yang disesuaikan panjang matrix sesuai dengan nilai sinyal yang dikirimkan setelah melakukan proses *cyclic prefix* yaitu sebesar 2560 simbol.

Tabel 3.10 Hasil Perhitungan kanal dan *multicarrier*

No	Kanal HST 2560x1
1	0,0058 – 0,1934i
2	0,0065 – 0,1911i
3	0,0072 – 0,1885i
…	….
2557	-4,5284e-4 + 2,2706e-5i
2558	-4,4626e-4 + 1,0211e-4i
2559	-5,3442e-4 – 9,8574e-5i
2560	-5,9501e-4 + 5,2596e-5i

3.14 *Remove Cyclic Prefix* Dan OFDM Demodulator (FFT)

Pada penerima, CP akan dihapus untuk mengembalikan data informasi asli. Penghapusan CP dilakukan dengan mengeliminasi sekitar 1/4 dari total simbol OFDM pada setiap *subcarrier* atau 512 simbol. Oleh karena itu simbol yang diterima pada blok OFDM demodulator sebesar 2048. Selanjutnya dilakukan proses *Fast Fourier Transform* (FFT) merupakan proses berkebalikan dari IFFT dimana merubah dari domain waktu menjadi domain frekuensi. Tujuan dari proses FFT adalah untuk memisahkan antara frekuensi pembawa dan simbol OFDM yang diterima pada penerima sebelum dilakukan demodulasi dan konversi kembali ke bentuk bit informasi. Proses FFT dapat diimplementasikan di MATLAB *simulink* dengan menggunakan blok OFDM Demodulator dan menghasilkan keluaran sebesar 1512 seperti pada Tabel 3.11.

Tabel 3.11 Menghapus *Cyclic Prefix* dan ofdm demodulator

No	Keluaran FFT 1512x1
1	-0,1316 – 0,0363i
2	-0,1330 – 0,0349i
3	-0,1355 – 0,0373i
…	…
1509	-0.1281 + 0.0697i
1510	-0.1214 + 0.0740i
1511	-0.1183 + 0.0767i
1512	-0.1134 + 0.0800i

3.15 *Symbol Demapping* 64-QAM

Pada proses demodulasi, sinyal keluaran dari FFT akan dipisahkan dari sinyal pembawa dan sinyal data, dan kemudian gelombang data yang tedeteksi akan dikembalikan ke dalam bentuk decimal seperti pada Tabel 3.12. Pada proses demodulasi setiap matriks menjadi 9072 simbol hal ini disebabkan perkalian 1512x6 menghasilkan 9072 simbol.

Tabel 3.12 Keluaran setelah *demapper*

No	Demapper 9072x1
1	1,7641
2	1,1470
3	-3,7641
…	…
9069	-3.4817
9070	-3.2652
9071	0.5183
9072	-0.7348

3.16 DVB – T *Inner Deinterleaver*

DVB – T *inner deinterleaver* merupakan proses yang dilakukan untuk mengembalikan urutan proses yang sudah di acak pada *inner interleaver* ke posisi semula. Saat proses pentransmisian sinyal, beberapa bit kemunkinan rusak atau terjadi *error* dalam kanal HST. *Inner deinterleaver* memilki peran dalam memperbaiki *error* yang terjadi pada saat pengiriman. Dengan mengembalikan urutan bit – bit data ke posisi semula, *deinterleaver* membantu memulihkan informasi yang hilang atau rusak akibat gangguan selama transmisi. Proses *inner deinterleaving* merupakan komponen penting pada sistem DVB – T yang berkontribusi dalam pemulihan *error* agar memberikan kualitas gambar suara yang lebih baik. Oleh karena itu bit yang dikirimkan akan dikembalikan seperti semula dapat dilihat pada Tabel 3.13.

Tabel 3.13 DVB – T *inner deinterleaver*

No	DVB-T Deinterleaver 2176x1
1	-1,1906
2	-0.1268
3	-2,8378
…	…
2173	0.5461

No	DVB-T Deinterleaver 2176x1
2174	0.2604
2175	-3.5853
2176	-1.0080

3.17 *Viterbi Decoder*

Decoder *Viterbi* menggunakan algorima *Viterbi* untuk mendekode *bitstream* yang telah dikodekan menggunakan *Forward Error Correction* (FEC) berdasarkan kode. Ada algoritma lain untuk mendekode aliran yang dikodekan secara konvolusional (misalnya, algoritma *Fano*). Algoritma *Viterbi* adalah yang paling memakan sumber daya, tetapi algoritma ini melakukan *decoding* dengan kemungkinan maksimum. Tabel 3.14 merupakan hasil decoding yang bernilai biner.

Tabel 3.14 Keluaran *viterbi decoder*

No	Viterbi Decoder 1632x1
1	1
2	0
3	1
…	…
1629	1
1630	0
1631	0
1632	0

3.18 *Deinterleaver*

Pada blok ini, melakukan proses *rearrange* bits. Pada sisi pemancar *interleaver* menghindari mungkin keluarnya dan *error fading* dengan merubah pesanan input bits. Oleh karena itu bit yang sudah dilakukan *interleaver* akan dikembalikan sepeti semula untuk dapat dilakukan *decoding* serta mengubah dalam bentuk desimal seperti pada Tabel 3.15.

Tabel 3.15 Keluaran *deinterleaver*

No	Deinterleaver 204x1
1	121
2	132
3	121
…	…
201	215
202	238
203	192

No	Deinterleaver 204x1
204	104

3.19 *Reed Solomon Decoding*

Pada blok ini simbol yang dikirimkan bersamaan dengan simbol *redundant* (simbol tambahan) akan dipisahkan. Jumlah bit tambahan yang digunakan pada penelitian ini yaitu 16 simbol (2t = 16). Dimana sebelumnya simbol yang dibawa 204 simbol menjadi 188 simbol dimana 8 bit untuk setiap simbolnya. Dapat dilihat pada Tabel 3.16 merupakan *receive* simbol pada sisi penerima.

Tabel 3.16 Keluaran *Reed Solomon decoding*

No	Reed Solomon Decoder 188x1
1	121
2	132
3	121
…	…
185	160
186	26
187	160
188	247