



UNIVERSITAS INDONESIA

**SIMULASI *BASEBAND POWER LINE COMMUNICATION*
(PLC) DENGAN MENGGUNAKAN DSK TMS320C6713
BERBASIS SIMULINK**

SKRIPSI

**MUHAMMAD FAROK KAMAJAYA
0405030508**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**



UNIVERSITAS INDONESIA

**SIMULASI *BASEBAND POWER LINE COMMUNICATION*
(PLC) DENGAN MENGGUNAKAN DSK TMS320C6713
BERBASIS SIMULINK**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

**MUHAMMAD FAROK KAMAJAYA
0405030508**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2010**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Muhammad Farok Kamajaya

NPM : 0405030508

Tanda Tangan : 

Tanggal : 23 Juni 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Muhammad Farok Kamajaya

NPM : 0405030508

Program Studi : Teknik Elektro

Judul Skripsi : Simulasi *Baseband Power Line Communication* (PLC) dengan
Menggunakan DSK TMS320C6713 Berbasis Simulink

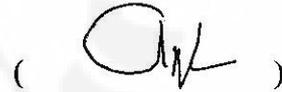
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Dadang Gunawan, M.Eng

()

Penguji : Dr. Ir. Arman D. Diponegoro

()

Penguji : Filbert Hilman Juwono S.T., M.T.

()

Ditetapkan di : Depok

Tanggal : 23 Juni 2010

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, sehingga saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat mata kuliah untuk menyelesaikan program studi di Departemen Teknik Elektro Universitas Indonesia.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Prof. Dr. Ir. Dadang Gunawan, M.Eng dan Dr. Ir. Arman D. Diponegoro, selaku dosen pembimbing 1 dan 2 yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Pihak laboratorium telekomunikasi Departemen Teknik Elektro FTUI yang telah banyak membantu dalam usaha memperoleh data dan peralatan yang saya perlukan;
- (3) Orang tua (M. Masrukhin dan Masrofah) dan kedua adik saya (Andi dan Indah) yang telah memberikan bantuan dukungan material dan moral. Semoga Allah SWT senantiasa melimpahkan rahmatNya kepada mereka; dan
- (4) Khotman Hilmy Fajrian, rekan satu bimbingan yang telah meminjamkan laptop untuk pengambilan data sekaligus mengajarkan penggunaan simulink pada DSK TMS320C6713; dan
- (5) Ajib Setyo Arifin, Ahmad Fauzi, Ahmad Saifullah, Maulana Ishak, Haris Setiawan, Daniel Ortega, Tedy, Aziz dan semua pihak yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu selanjutnya.

Depok, 23 Juni 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Farok Kamajaya

NPM : 0405030508

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

**Simulasi *Baseband Power Line Communication* (PLC) dengan Menggunakan
DSK TMS320C6713 Berbasis Simulink**

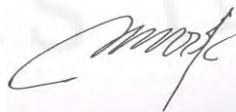
beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 23 Juni 2010

Yang menyatakan



(Muhammad Farok Kamajaya)

ABSTRAK

Nama : Muhammad Farok Kamajaya
Program Studi : Teknik Elektro
Judul : Simulasi *Baseband Power Line Communication* (PLC) dengan Menggunakan DSK TMS320C6713 Berbasis Simulink

Skripsi ini membahas sebuah rancangan simulasi *baseband Power Line Communication* (PLC) dalam sistem komunikasi digital dengan menggunakan program Simulink. Parameter yang digunakan pada simulasi ini mengikuti standar homeplug 1.0 yang menggunakan OFDM dan penyandian FEC. Model simulasi *baseband* PLC ini kemudian diuji pada Matlab Simulink dan DSP prosesor pada perangkat DSK TMS320C6713 berbasis simulink. Dari hasil pengujian diperoleh nilai data keluaran yang sama, baik pada pengujian menggunakan simulink maupun pengujian pada DSK TMS320C6713 dengan RTDX. Hal ini menunjukkan bahwa model simulasi yang telah dibuat telah berhasil diterapkan pada papan DSK TMS320C6713.

Kata kunci : PLC, OFDM, DQPSK, Reed-Solomon, Kode Konvolusi, Simulink, RTDX, DSK TMS320C6713

ABSTRACT

Name : Muhammad Farok Kamajaya
Study Program : Electrical Engineering
Title : Baseband Power Line Communication (PLC) Simulation Using DSK TMS320C6713 Based On Simulink

This thesis discusses a design of a baseband Power Line Communication (PLC) simulation in a digital communication system using Simulink program. The parameters of simulation refers to homeplug 1.0 standard that uses Orthogonal Frequency Division Multiplexing (OFDM) and FEC encoding. The Baseband PLC model design then being tested in Matlab Simulink and DSP processors on the device DSK TMS320C6713 based on simulink. From the test results obtained by the same amount of output data, both in testing using simulink and DSK TMS320C6713 with RTDX. This result show that the simulation model has been successfully applied to the DSK TMS320C6713 board.

Keyword : PLC, OFDM, DQPSK, Reed-Solomon, Convolutional, Simulink, RTDX, DSK TMS320C6713

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
LEMBAR PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR SINGKATAN	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian.....	3
1.5 Sistematika Penulisan	3
BAB 2 BASEBAND PLC, SIMULINK DAN DSK TMS 320C6713	4
2.1 <i>Power Line Communication</i> (PLC)	4
2.2 <i>Baseband</i> PLC	5
2.2.1 Teknik Pengkodean FEC.....	5
2.2.1.1 Reed Solomon.....	6
2.2.1.2 Kode Konvolusi	7
2.2.2 <i>Orthogonal Frequency Division Multiplexing</i> (OFDM)	8
2.2.2.1 Prinsip Kerja OFDM.....	10
2.2.2.2 Keuntungan OFDM.....	12
2.2.3 <i>Differential Quadrature Phase Shift Keying</i> (DQPSK)	12
2.3 Simulink	13
2.4 DSK TMS320C6713	15
BAB 3 PERANCANGAN SIMULASI RANGKAIAN BASEBAND PLC ...	18
3.1 Perancangan Model Baseband PLC dengan Simulink	18
3.1.1 Diagram Blok <i>Transmitter</i>	18
3.1.1.1 Blok <i>Input data</i>	18
3.1.1.2 Blok <i>Scrambler</i>	20
3.1.1.3 Blok <i>Reed Solomon Encoder</i>	22
3.1.1.4 Blok <i>Convolutional Encoder</i>	22
3.1.1.5 Blok <i>Puncture</i>	24
3.1.1.6 Blok <i>Interleaver</i>	25
3.1.1.7 Blok <i>IQ Mapper</i>	25
3.1.1.8 Blok OFDM Simbol	25
3.1.2 Kanal <i>noise</i> PLC.....	27
3.1.3 Diagram Blok Receiver.....	27
3.2 Penerapan Model Simulink pada DSK TMS320C6713	27

BAB 4 UJI COBA DAN ANALISIS HASIL SIMULASI.....	34
4.1 Hasil Pengujian dengan Simulink	34
4.1.1 Hasil Data <i>Input</i> dan Data <i>Output</i>	34
4.1.2 Perbandingan Sinyal dan Spektrum OFDM.....	39
4.2 Hasil Pengujian dengan DSK TMS320C6713	41
4.2.1 Hasil Data <i>Output</i>	41
4.3 Perbandingan Hasil Simulink dan Hasil DSK TMS320C6713	41
BAB 5 KESIMPULAN	43
DAFTAR REFERENSI	44
DAFTAR PUSTAKA	45
LAMPIRAN.....	47

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Blok diagram PLC.....	4
Gambar 2.2 Blok diagram <i>baseband</i> PLC sisi pengirim.....	5
Gambar 2.3 <i>Multicarrier</i> OFDM	8
Gambar 2.4 Spektrum frekuensi pada FDM	8
Gambar 2.5 Spektrum OFDM	9
Gambar 2.6 OFDM <i>multicarrier technique</i>	9
Gambar.2.7. Blok dasar OFDM	11
Gambar 2.8. $\pi/4$ DQPSK dalam konstelasi.....	13
Gambar 2.9. <i>Simulink library browser</i>	14
Gambar 2.10 Lembar kerja Simulink	15
Gambar 2.11 Bentuk fisik DSK TMS320C6713	16
Gambar 2.12 Blok diagram DSK TMS320C6713	16
Gambar 3.1 Model Simulasi rangkaian <i>baseband</i> PLC pada simulink	18
Gambar 3.2 Parameter Blok <i>Sample Data</i>	19
Gambar 3.3 Parameter Blok <i>PN Sequence Generator</i>	21
Gambar 3.4 Parameter Blok <i>Reed Solomon</i>	22
Gambar 3.5 Parameter Blok <i>Convolutional Encoder</i>	23
Gambar 3.6 Bagan ilustrasi rangkaian <i>convolutional encoder</i> $\frac{1}{2}$	24
Gambar 3.7 Parameter blok <i>puncture</i>	24
Gambar 3.8 Blok diagram pada <i>IQ Mapper</i>	25
Gambar 3.9 Blok OFDM Modulation.....	26
Gambar 3.10 Model Simulasi Dengan <i>DSP Board</i>	28
Gambar 3.11 Konfigurasi parameter Model	29
Gambar 3.12 Proses <i>Diagnostic</i>	30
Gambar 3.13 Diagram Alir Targetting to C6000 DSP.....	30
Gambar 3.14 Proses pembuatan porgram C.....	31

Gambar 3.15 Proses Load Program	32
Gambar 3.16 Model untuk plot data keluaran RTDX.....	33
Gambar 4.1 Pencantuman blok scope pada model simulink.....	34
Gambar 4.2 Data <i>input integer</i>	36
Gambar 4.3 Data <i>output integer</i>	36
Gambar 4.4 Perbandingan data bit kirim (atas) dan data bit terima (bawah)	37
Gambar 4.5 Pembuktian Data	38
Gambar 4.6 Perbandingan Sinyal OFDM kirim dan OFDM terima.....	39
Gambar 4.7 Spektrum OFDM sisi pengirim.....	40
Gambar 4.8 Spektrum OFDM sisi penerima.....	40
Gambar 4.9 Data Output RTDX dalam bit	41
Gambar 4.10 Data Output RTDX dalam integer	42

DAFTAR TABEL

	Halaman
Tabel 3.1 Blok XOR	21
Tabel 3.2 Parameter OFDM.....	26
Tabel 4.1 Variabel “data input”	35

DAFTAR SINGKATAN

AWGN	<i>Additive White Gaussian Noise</i>
BER	<i>Bit Error Rate</i>
CCS	<i>Code Composer Studio</i>
DFT	<i>Discrete Fourier Transform</i>
DSK	<i>Digital Signal Processing Starter Kit</i>
DSP	<i>Digital Signal Processing</i>
DQPSK	<i>Differential Quadrature Phase Shift Keying</i>
FEC	<i>Forward Error Correction</i>
FDM	<i>Frequency Division Multiplexing</i>
GF	<i>Galois Field</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
FFT	<i>Fast Fourier Transform</i>
IDFT	<i>Inverse Discrete Fourier Transform</i>
IFFT	<i>Inverse Fast Fourier Transform</i>
PLC	<i>Power Line Communication</i>
RTDX	<i>Real Time Data Exchange</i>
SNR	<i>Signal to Noise Ratio</i>
TI	<i>Texas Instrument</i>
USB	<i>Universal Serial Bus</i>
XOR	<i>Exclusive OR</i>

BAB I PENDAHULUAN

1.1. Latar Belakang

Teknologi informasi dan komunikasi (TIK) kini menjadi kebutuhan masyarakat luas tidak hanya di perkotaan. Namun terbatasnya pelayanan dan infrastruktur pada pedesaan menyebabkan layanan TIK tidak dapat dinikmati secara merata. Salah satu cara untuk mengatasinya yaitu dengan memanfaatkan teknologi PLC yang menggunakan jaringan listrik pada perumahan. Infrastruktur jaringan listrik umumnya lebih tersedia dan menjangkau pedesaan bila dibandingkan dengan jaringan telepon, *internet* atau *wireless*.

Power Line Communication (PLC) merupakan sistem transmisi data yang menggunakan media jalur listrik sebagai pembawa frekuensi carrier. Namun propagasi sinyal melalui jalur listrik ini memiliki kelemahan, yaitu banyak redaman, *noise* dan interferensi medan elektromagnetik yang dapat menurunkan kualitas informasi hingga hilangnya informasi data yang dikirim[2]. Untuk itulah diperlukan teknik transmisi, pengkodean kanal dan modulasi yang tepat untuk memperbaiki kinerja sistem dalam mentransmisikan data.

Orthogonal Frequency Division Multiplexing (OFDM) merupakan gabungan dari teknik transmisi dan modulasi yang memungkinkan spektrum sinyalnya dibagi-bagi ke dalam beberapa *frequency subcarrier* yang saling tegak lurus (*orthogonal*). Karakteristik yang saling tegak lurus membuat *frequency subcarrier* dapat saling *overlap* tanpa menimbulkan interferensi satu dengan yang lain sehingga sangat efisien dalam penggunaan *bandwidth*. Sedangkan teknik pengkodean yang digunakan adalah *Reed Solomon* dan *Convolutional Encoder* yang merupakan bentuk penyandian dengan teknik *forward error correction* (FEC) yaitu kemampuan untuk mendeteksi dan mengoreksi adanya kesalahan data yang diterima dengan bantuan bit-bit *redundant*. Penggunaan kedua teknik pengkodean FEC tersebut diharapkan dapat meningkatkan kehandalan sistem dalam memperbaiki kesalahan yang lebih bervariasi. Teknik OFDM dan pengkodean FEC merupakan komponen utama penyusun *baseband* pada PLC ini.

Perancangan model simulasi *baseband Power Line Communication* (PLC) dibuat dengan menggunakan simulink. Setelah model berhasil dibuat dan diatur konfigurasinya baru kemudian diterapkan ke dalam prosesor DSP (*Digital Signal Processors*) pada *Digital Signal Processing Starter Kit* (DSK) TMS320C6713. Penerapan model pada DSK tidak membutuhkan pemrograman ulang karena Simulink akan membuat kode bahasa pemrograman C untuk dapat diterapkan dalam DSK. Kode bahasa pemrograman ini akan dijalankan dalam perangkat lunak *Code Composer Studio* (CCS) dan dari perangkat lunak ini kemudian program dijalankan pada DSK. Dengan fasilitas RTDX pada simulink maka data keluaran dari proses *targetting* ke DSK dapat disimpan dalam *workspace*. Data keluaran dari RTDX ini yang akan dibandingkan dengan data pada simulink untuk menguji kebenarannya. Apabila hasil yang didapat sesuai maka model Simulink yang dibuat telah berhasil diterapkan pada DSK.

1.2. Perumusan Masalah

Berdasarkan latar belakang tersebut, dapat dirumuskan masalah yaitu bagaimana merancang simulasi rangkaian *baseband* PLC agar dapat diterapkan pada perangkat DSP prosesor. Rumusan masalah dapat diperinci menjadi tiga pertanyaan sebagai berikut:

1. Bagaimana merancang model simulasi rangkaian *baseband* PLC dengan menggunakan program simulink?
2. Bagaimana model rangkaian *baseband* PLC dapat berjalan pada simulink tanpa mengalami *error*?
3. Bagaimana menerapkan rancangan dari program simulink ke dalam perangkat DSK TMS320C6713?

1.3. Tujuan Penulisan

Tujuan skripsi ini adalah untuk merancang model simulasi dari *baseband PLC* pada DSK TMS320C6713 dengan berbasis Simulink. Sehingga model yang dibuat dapat berjalan pada DSK dengan hasil data uji yang sama seperti pada hasil pengujian Simulink.

1.4. Batasan Masalah

Adapun batasan masalah pada skripsi ini adalah:

1. Tidak melakukan ujicoba pada berbagai jenis *input*.
2. Tidak melakukan ujicoba dengan berbagai macam jenis gangguan.
3. Tidak melakukan ujicoba pada osiloskop.

1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan pada skripsi ini yaitu:

1. Studi literatur mengenai *baseband Power Line Communication* (PLC) dan materi pendukung skripsi lainnya.
2. Merancang model *baseband* PLC dan algoritmanya dalam bentuk blok diagram model simulasi dengan menggunakan perangkat lunak Matlab Simulink R2007a.
3. Penerapan pada perangkat keras DSK TMS320C6713 yang dilakukan dengan bantuan perangkat lunak tambahan yaitu *Code Composer Studio* (CSS).

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Menjelaskan tentang latar belakang dan dasar pemikiran *baseband* PLC, tujuan penulisan, batasan masalah dalam pembahasan tugas skripsi ini, beserta metodologi penelitian yang digunakan dan sistematika penulisan.

BAB II BASEBAND PLC, SIMULINK DAN DSK TMS320C6713

Menjelaskan tentang pengetahuan-pengetahuan dasar yang berkaitan dengan *baseband* PLC. Serta pengenalan perangkat lunak (Simulink, CCS) dan perangkat keras (DSK TMS320C6713) yang digunakan dalam pembuatan skripsi ini.

BAB III PERANCANGAN SIMULASI *BASEBAND* PLC

Menjelaskan tentang perancangan simulasi *baseband* PLC pada perangkat lunak Matlab Simulink R2007a serta penerapannya pada perangkat keras DSK TMS320C6713.

BAB IV PENGUJIAN DAN ANALISIS SISTEM *BASEBAND* PLC

Menjelaskan tentang uji coba model simulasi, pengambilan data dan menganalisa hasil simulasi dari model *baseband* PLC yang telah dibuat pada Simulink dan diterapkan pada DSK TMS320C6713.

BAB V KESIMPULAN

Memberikan kesimpulan dari keseluruhan skripsi ini.

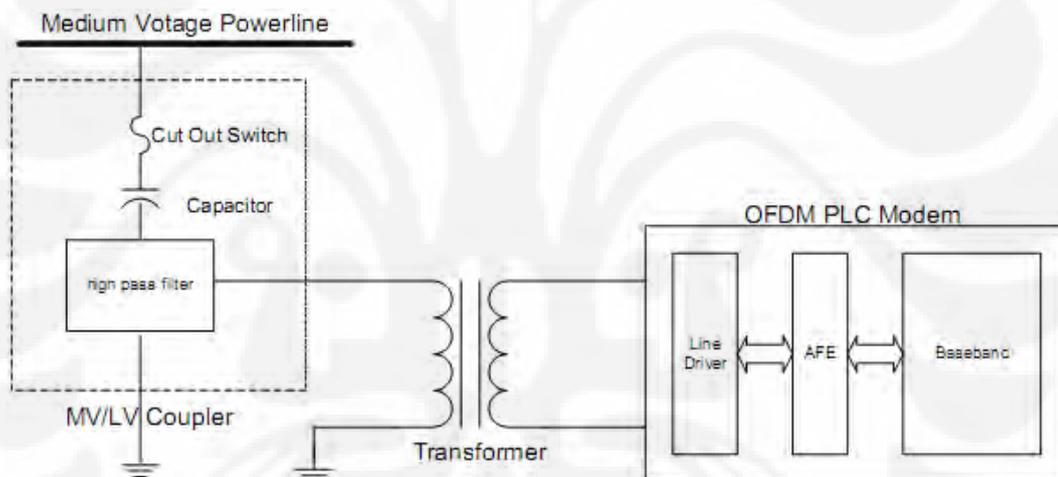
BAB II

BASEBAND PLC, SIMULINK DAN DSK TMS 320C6173

2.1. Power Line Communication (PLC)

Teori ini diambil dari landasan teori: "Simulasi Baseband PLC dengan menggunakan Simulink" oleh M. Farok Kamajaya.

Power Line Communication (PLC) adalah komunikasi data yang dilakukan melalui jalur listrik. Dasar kerja PLC adalah menggunakan frekuensi tinggi yang tidak digunakan untuk mengalirkan listrik. Jalur listrik umumnya menggunakan frekuensi 50-60 Hz untuk mengalirkan daya listrik. PLC akan menggunakan frekuensi yang lebih tinggi sebagai media pembawa (*carrier*) untuk mengirimkan data [1]. Gambar 2.1 menunjukkan blok diagram PLC.



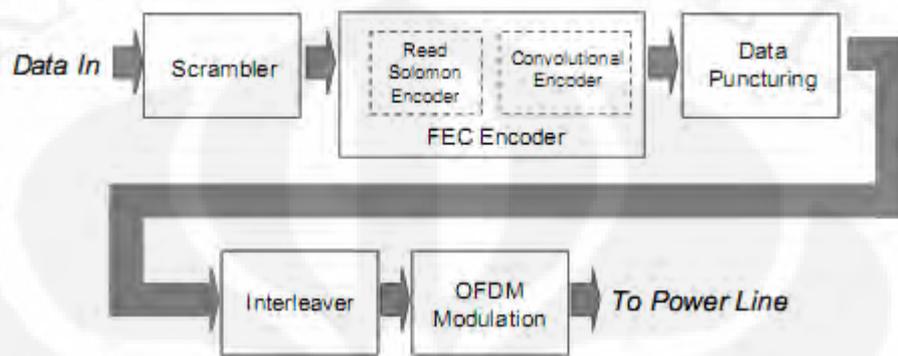
Gambar 2.1 Blok diagram PLC [3]

Pada Gambar 2.1, data yang dikirim berupa data digital yang mengalami penyandian dan modulasi pada blok *baseband*. Selanjutnya, data melalui blok AFE (*Analog Front End*) yang akan mengubah sinyal digital menjadi analog (sinusoidal). Kemudian sinyal akan mengalami proses *filtering* dan perubahan frekuensi agar data dapat melalui jaringan listrik. Blok *Line Driver* akan meneruskan pada transformator yang akan mengatur tegangan keluaran sesuai pada jalur listrik. Selanjutnya *MV/LV Coupler* akan menuju gardu listrik bertegangan menengah/bertegangan rendah (*medium voltage/low voltage*) yang berasal dari perumahan atau perkantoran.

2.2. Baseband PLC

Teori ini diambil dari landasan teori: "Simulasi *Baseband* PLC dengan menggunakan Simulink" oleh M. Farok Kamajaya.

Pada skripsi ini akan dibuat model dari *baseband* PLC. Gambar 2.2 adalah blok diagram *baseband* PLC dari standar *homeplug* 1.0 pada sisi pengirim.



Gambar 2.2 Blok diagram *baseband* PLC sisi pengirim [2]

Blok-blok diagram pada Gambar 2.2 adalah komponen penyusun *baseband* PLC. Berikut ini akan dibahas teknologi penyusun *baseband* PLC

2.2.1. Teknik Pengkodean FEC

Pengkodean merupakan cara untuk meningkatkan performa sistem. Dimana jika terjadi kesalahan maka sistem akan berusaha memperbaiki cacat tersebut. Cara perbaikan ini dapat menggunakan teknik retransmisi dan teknik FEC (*Forward Error Correction*). Untuk teknik retransmisi, data yang rusak akan diminta untuk dikirim kembali, sehingga boros daya, *bandwidth* dan waktu. Sedangkan teknik FEC, memperbaiki data yang rusak pada sisi penerima dengan bantuan dari *bit-bit redundant*.

FEC merupakan fitur umum dari semua transmisi digital di mana kesalahan dapat dikoreksi oleh redundansi informasi yang dibawa dalam sinyal yang dikirimkan itu sendiri. Hal ini untuk menghindari kebutuhan untuk mengulang *frame* dan memelihara *throughput*. Bit yang berlebihan dapat dilakukan melalui berbagai cara mulai dari paritas bit ke repetisi sederhana yang dikirimkan sinyal. Namun, untuk efisiensi maksimum, bit FEC harus disesuaikan dengan karakteristik dari sistem transmisi. Convolutional coding dengan Reed Solomon (RS) merupakan penyandian FEC yang memperbaiki kesalahan kode pada PLC.

2.2.1.1 Reed Solomon (RS)

Kode Reed Solomon berfungsi untuk mengatasi kesalahan yang terjadi pada *bursts*. Prinsip kerja dari kode reed solomon yaitu dengan membuat terlebih dahulu suatu *polynomial* dari suatu simbol data untuk dipancarkan dan kemudian mengirimkan suatu versi yang *oversampled* dari *polynomial* tersebut sebagai pengganti simbol asli dari data tersebut.

➤ Penyandian *Reed Solomon*

Suatu kode *reed solomon* dilambangkan sebagai RS (n,k,t) dengan simbol i-bit. Maksudnya adalah bahwa *encoder* memberikan k data simbol pada masing-masing i-bit dan menambahkan 2t simbol *parity* untuk memperbaiki n-simbol *codeword*. Jadi n,k,t dapat didefinisikan sebagai berikut:

- n : nomor dari bytes setelah penyandian.
- k : nomor dari data bytes sebelum penyandian.
- t : nomor dari data bytes yang dapat diperbaiki

Kemampuan koreksi kesalahan dari sebuah kode RS ditentukan oleh (n-k). Jika letak simbol yang salah tidak diketahui, maka kode RS dapat mengoreksi sampai t simbol, dimana t dilambangkan sebagai $t = (n-k)/2$

Pada *reed solomon* terdapat *Primitive Polynomial* dan *generator polynomial*. *Primitive Polynomial* digunakan untuk membuat *symbol field* dan disebut juga sebagai suatu *symbol field* dari *generator polynomial*. Kode *generator polynomial* digunakan untuk menghitung simbol *parity* dan mempunyai spesifikasi format yg ditetapkan, yakni[14]:

$$g(x) = (x+\lambda^0)(x+\lambda^1)(x+\lambda^2)\dots(x+\lambda^{2t-1}) \quad (2.1)$$

Di mana λ adalah elemen *parity* dari *Galois field* di mana pesan masukannya telah didefinisikan.

➤ Penguraian Reed Solomon

Pengurai *Reed Solomon* melaksanakan operasi yang diperlukan untuk memecahkan kode sinyal dan pada bagian akhir mendapatkan pesan yang asli yang dikirim dari sumber. Seperti pada blok semua penerima, Pengurai *Reed Solomon* membalikkan langkah-langkah yang terjadi pada penyandian. Pengurai *Reed Solomon* bekerja dengan mengambil *codewords* yang panjangnya n setelah

sinyal dipecahkan dan lalu mengembalikan pesan yang panjangnya k , prosesnya seperti RS encoder yang membedakan adalah prosesnya yang berkebalikan.

2.2.1.2 Kode Konvolusi

Kode konvolusi berfungsi untuk menurunkan *error rate* selama proses pentransmisi data, atau dengan kata lain berfungsi sebagai pendeteksi dan sekaligus memperbaiki error selama pentransmisi data, tanpa harus meminta pengirim untuk mengirimkan kembali data yang sama. Prinsip kerja dari kode konvolusi adalah dengan men-xor-kan beberapa bit masukan sebelumnya dengan bit input pada saat itu. Sistem xor ini mencakup bagaimana alur operasi xor dan berapa lama suatu bit akan mempengaruhi keluaran bit lain.

➤ *Convolutional Encoder*

Convolutional encoder adalah sebuah blok yang berfungsi untuk mengkodekan urutan dari input vektor biner untuk menghasilkan output vektor biner. Code convolutional dispesifikasikan dalam 3 parameter (n, k , dan m)

- n = jumlah bit keluaran
- k = jumlah bit masukan
- m = jumlah memori register

Kuantitas dari kode k/n disebut *code rate* yang merupakan acuan tingkat efisiensi dari kode. Umumnya parameter k dan n berkisar antara 1 sampai 8, m berada diantara 2 sampai 10 sedangkan code rate dari encoder antara $1/8$ sampai $7/8$. Umumnya kode konvolusi dinyatakan dengan parameter dalam variabel (n, k, K) dengan nilai K :

$$\text{Constrain Length, } K = k(m-1) \quad (2.2)$$

Suatu pengkodean konvolusi menerima pesan yang panjangnya k dan menghasilkan *codeword* n bit. Secara umum, hal itu terdiri dari suatu shift register L segmen, di mana L menandakan panjangnya batasan tersebut. Jika terjadi suatu garis koneksi dari daftar pergeseran, maka diwakilkan oleh sebuah bit “satu” di oktal penyajian dari polinomial dan jika tidak ada koneksi diwakili oleh bit “nol”.

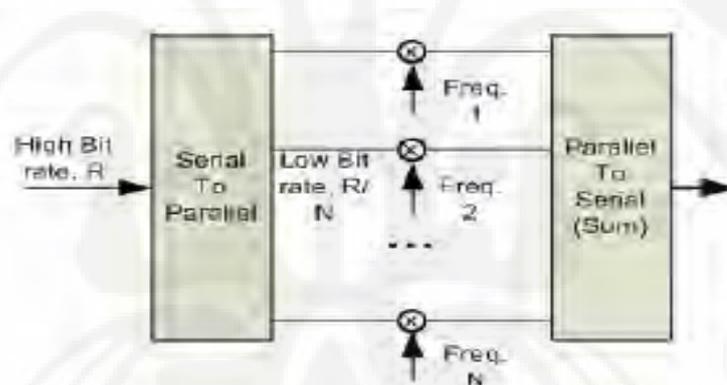
➤ *Viterbi Decoder*

Decoding yang digunakan adalah *Viterbi decoding*. *Viterbi decoding* merupakan pasangan *encode-decode* dari *Convolutional encoding* yang menggunakan algoritma *Viterbi* dalam melakukan *decoding* data. Untuk dapat

mengekstrak data dengan benar, tentunya parameter pada *Viterbi decoding* harus sama dengan parameter pada *Convolutional Encoding*, seperti sistem *xor* dan *coding rate*. Karena *decoding* merupakan sistem *reversible* dari *encoding*.

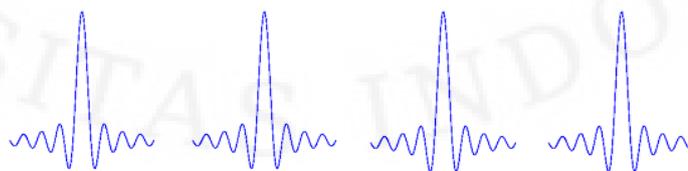
2.2.2. Orthogonal Frequency Division Multiplexing (OFDM)

OFDM pada dasarnya merupakan suatu skema transmisi *multi carrier* yang merupakan kombinasi dari modulasi dan *multiplexing*. *Multiplexing* adalah metode untuk menggunakan *bandwidth* secara bersamaan untuk beberapa kanal data yang berbeda. OFDM adalah teknik transmisi yang berkembang dari teknik FDM. Perbedaan utama keduanya terletak dari segi efisiensi pemakaian *bandwidth*. Perbedaan tersebut disebabkan cara kerja OFDM yang lebih baik dibandingkan FDM. Pada OFDM, informasi dengan *high data rate* dibagi menjadi beberapa bagian informasi *low data rate* dan menggunakan *frequency subcarrier* yang saling tegak lurus (*orthogonal*) antara yang satu dengan yang lain seperti yang dapat dilihat pada Gambar 2.3.



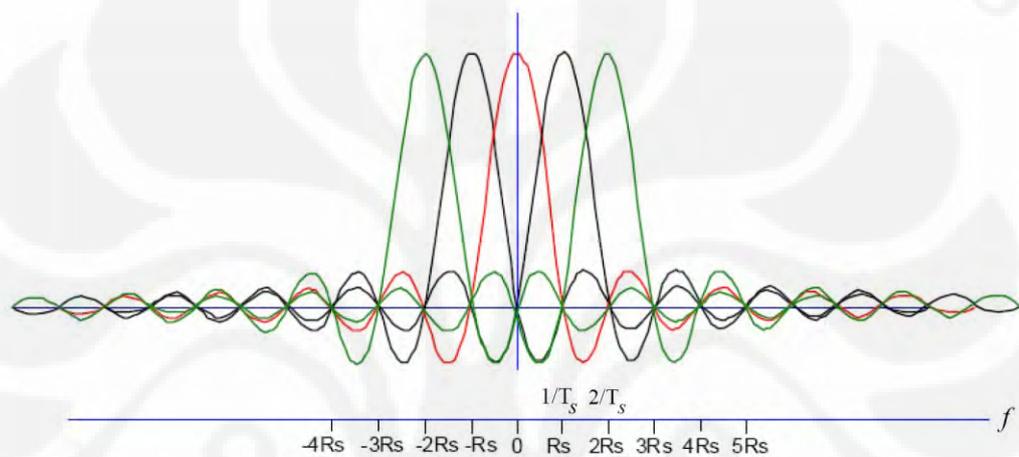
Gambar 2.3 Multicarrier OFDM[6]

Konsep dasar OFDM berbeda dengan FDM (*frequency Division Multiplexing*). Pada FDM, suatu *bandwidth* tertentu dibagi menjadi beberapa kanal tersendiri berdasarkan frekuensi. Agar masing-masing kanal tidak saling menginterferensi satu sama lain maka diberi jarak antar kanal (*guardband*) seperti yang ditunjukkan pada Gambar 2.4. Hal ini tentunya akan membuat penggunaan *bandwidth* tidak efisien.



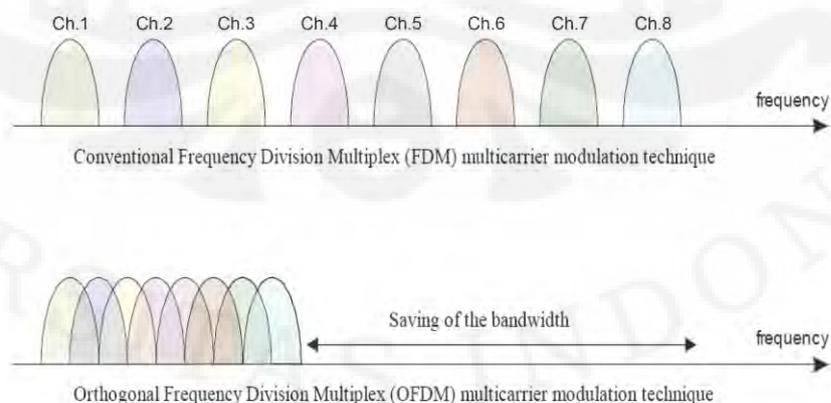
Gambar 2.4 Spektrum frekuensi pada FDM[6]

Berbeda halnya dengan FDM, pada OFDM penggunaan *bandwidth* sangat efisien. Keempat kanal yang ada dalam satu *bandwidth* seakan-akan ditumpang tindihkan menjadi satu, namun tidak terjadi saling interferensi antar kanal. Hal ini dikarenakan masing-masing sinyal transmisi dalam setiap kanal bersifat saling *orthogonal* dan *harmonic* satu sama lain. *Orthogonal* pada OFDM memiliki arti bahwa spektrum frekuensi dari suatu *subcarrier* bernilai nol saat spektrum frekuensi *subcarrier* yang bersebelahan bernilai maksimum seperti digambarkan pada Gambar 2.5.



Gambar 2.5 Spektrum OFDM [6]

Overlap yang terdapat pada OFDM menjadikan proses pengiriman data menjadi lebih cepat dan tingkat efisiensi pemakaian *bandwidth* yang sangat tinggi seperti yang dapat dilihat pada Gambar 2.6.



Gambar 2.6 OFDM *multicarrier technique*

Seperti yang terlihat pada Gambar 2.6, sinyal terbagi menjadi *subcarrier* yang *orthogonal* sehingga sinyal yang terbentuk adalah sinyal *narrowband* (beberapa kHz) dan oleh karena itu sinyal ini sangat tahan terhadap efek *multipath*. OFDM juga menyediakan frekuensi gain sehingga dapat menambah performansi pada *physical layer*.

2.2.2.1. Prinsip Kerja OFDM

Deretan data informasi yang akan dikirim dikonversikan ke bentuk paralel, sehingga bila *bit rate* semula adalah R, maka *bit rate* di tiap-tiap jalur paralel adalah R/M. M adalah jumlah jalur paralel (sama dengan jumlah *subcarrier*). Setelah itu, modulasi dilakukan pada tiap-tiap *subcarrier*. Modulasi bisa berupa BPSK, QPSK, QAM atau yang lain. BPSK, QPSK, dan QAM adalah beberapa teknik modulasi yang sering digunakan pada OFDM. Pada skripsi ini digunakan modulasi DQPSK.

Sinyal yang telah termodulasi tersebut kemudian diaplikasikan ke *Inverse Discrete Fourier Transform* (IDFT) untuk pembuatan simbol OFDM dan mengubah *frequency domain* menjadi *time domain*. Setelah itu, simbol-simbol OFDM dikonversikan lagi kedalam bentuk serial kemudian sinyal dikirim. Sinyal yang dikirim dapat dinyatakan melalui persamaan (2.3)[7]:

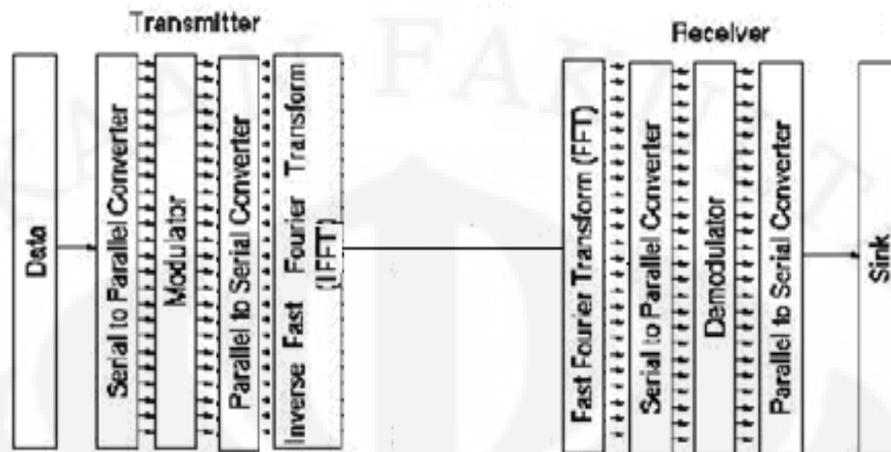
$$s(t) = \text{Re} \left\{ \sum_{n=-\infty}^{+\infty} b_n f(t-nT) e^{j(\omega_n t + \phi)} \right\} \quad (2.3)$$

Penggunaan IDFT ini memungkinkan pengalokasian frekuensi yang saling tegak lurus (*orthogonal*) dan mengubah domain sinyal dari *frequency domain* ke *time domain* untuk selanjutnya sinyal dikirim ke *receiver* OFDM. Penerapan IDFT dapat dinyatakan melalui persamaan (2.4)[7]:

$$X(nT) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp(jk2\pi \frac{n}{N}) \quad (2.4)$$

Sinyal OFDM dibangkitkan dan diproses secara digital. Hal ini dimaksudkan untuk mengurangi kerumitan menyediakan sejumlah besar osilator. Oleh karena itu, proses modulasi dan demodulasi dilakukan dengan teknik

pengolahan digital, yakni *Discrete Fourier Transform* (DFT). Blok dasar OFDM yang lengkap dapat dilihat pada Gambar 2.7.



Gambar 2.7 Blok dasar OFDM [7]

Teknologi *Orthogonal Frequency Division Multiplexing* (OFDM) menggunakan beberapa carrier (*multicarrer*) dalam bandwidth yang telah dialokasikan, masing-masing carrier membawa datarate yang lebih rendah dari single carrier. Teknologi penggunaan multicarrer akan membuat signal menjadi tahan terhadap kanal multipath. Bagaimanapun juga dalam implementasinya, harus ada hubungan yang spesial antara masing-masing subcarrier tersebut, hubungan tersebut disebut dengan *Orthogonal*. Suatu carrier dikatakan orthogonal dengan carrier yang lainnya apabila (pada domain frekuensi) sideband *null* jatuh tepat pada main lobe frekuensi carrier selanjutnya.

Secara matematis dua buah signal dikatakan orthogonal apabila[7]:

- a. Untuk signal kontiyu

$$\int_0^{T_s} \cos(2\pi n f_0 t) \times \cos(2\pi m f_0 t) dt = 0 ; n \neq m \quad (2.5)$$

- b. Untuk signal diskrit

$$I \sum_{k=0}^{N-1} \cos\left(\frac{2\pi kn}{N}\right) \times \cos\left(\frac{2\pi km}{N}\right) = 0 ; n \neq m \quad (2.6)$$

Dimana T_s adalah periode symbol dan N adalah jumlah subcarriernya. Besarnya frekuensi subcarrier yang digunakan dapat dinyatakan sebagai :

$$f_k = f_0 + \frac{k}{T_S} ; k = 0, 1, 2, \dots, N - 1 \quad (2.7)$$

Dari persamaan (2.7) diatas dapat diperoleh jarak setiap frekuensi subcarrier agar saling orthogonal yang dinyatakan dalam persamaan (2.8):

$$\Delta f = \frac{1}{T_S} \quad (2.8)$$

Dimana Δf adalah jarak antara frekuensi subcarrier.

2.2.2.2 Keuntungan OFDM

Keuntungan menggunakan OFDM dalam sistem PLC, yaitu[5]:

- Efisiensi *bandwidth*
- Memiliki ketahanan terhadap *noise*
- Interferensi antara sub-band tidak terjadi karena masing-masing sinyal carrier saling orthogonal satu sama lain
- Fleksible dan dapat dibuat adaptif terhadap kondisi kanal
- Skalabilitas untuk *data rate* yang tinggi

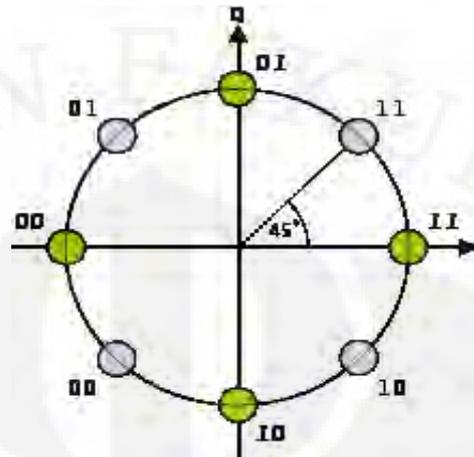
2.2.3 DQPSK (*Differential Quadrature Phase Shift Keying*).

Teori ini diambil dari landasan teori:”Simulasi *Baseband* PLC dengan menggunakan Simulink” oleh M. Farok Kamajaya.

DQPSK merupakan variasi dari QPSK, yaitu suatu teknik pengkodean *M-ary* dimana $M=4$. *M-ary* adalah suatu bentuk turunan dari kata *binary*. *M* berarti digit yang mewakili banyaknya kondisi yang mungkin. Dalam QPSK terdapat empat keluaran fasa yang berbeda begitu juga dengan inputnya. Karena input digital ke modulator QPSK adalah sinyal biner, maka untuk menghasilkan empat kondisi input yang berbeda harus dipakai bit input lebih dari satu bit tunggal. Dengan menggunakan 2 bit, maka terdapat empat kondisi yang mungkin, yaitu 00, 01, 10 dan 11. Karena mengkodekan dua bit dalam satu symbol, maka *data rate* pada QPSK sama dengan dua kali *data rate* pada BPSK. Pada QPSK, bit-bit genap atau ganjil digunakan untuk memodulasi komponen-komponen *in-phase* dan *quadrature* dari sinyal *carrier*.

DQPSK adalah sebuah teknik modulasi digital dimana pengiriman data dilakukan dengan perubahan secara modulasi fasa sinyal pembawanya. Pada DQPSK, yang memodulasi 2 bit informasi setiap simbolnya dengan mengubah

fasa rotasi $\pi/4$ QPSK. Konstelasi dari DQPSK ditunjukkan pada Gambar 2.8 dibawah ini:

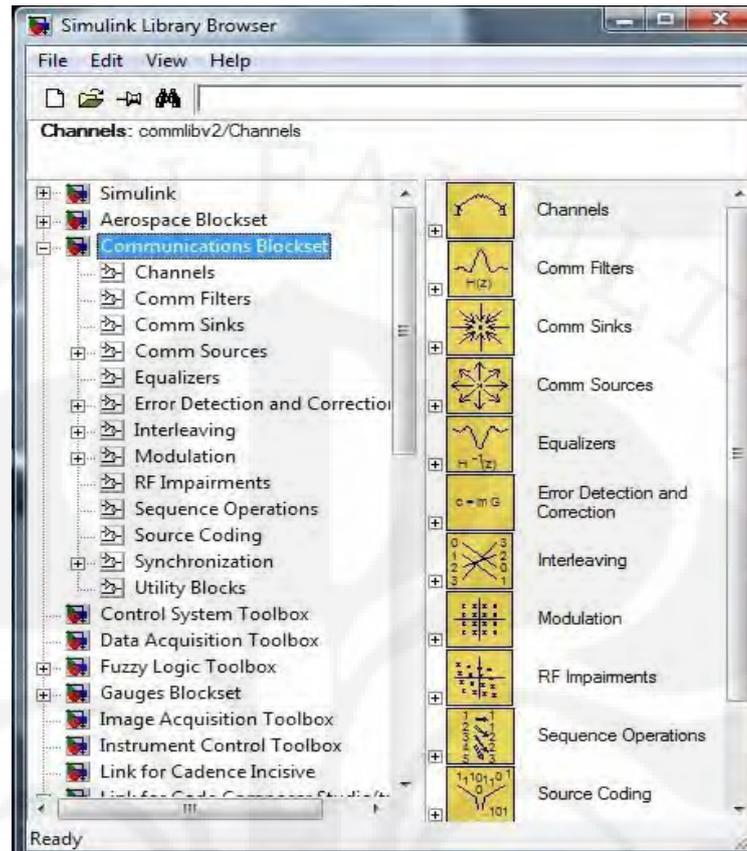


Gambar.2.8. $\pi/4$ DQPSK dalam konstelasi[11]

2.3 Simulink

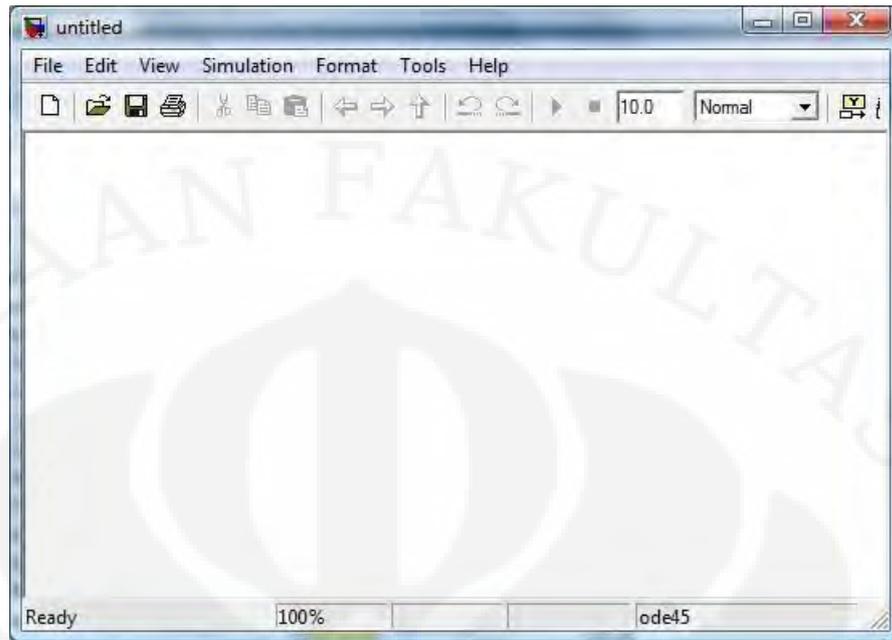
Teori ini diambil dari landasan teori:”Simulasi *Baseband* PLC dengan menggunakan Simulink” oleh M. Farok Kamajaya.

Simulink yang dikembangkan oleh The MathWorks, adalah *sub-program* dari MATLAB yang digunakan untuk simulasi, pemodelan atau analisis sistem. Simulink dapat melakukan berbagai macam simulasi dengan cepat dan mudah. Bahkan setelah penambahan beberapa program yang bersifat *Real Time Workshop* (contoh: Code Composer Studio), simulink mampu menghasilkan bahasa pemrograman C dari model yang telah dibuat. Simulink dapat dijalankan dari menu utama, yaitu dengan mengetik “simulink” atau dengan menekan icon yang menandakan simulink, setelah itu, user akan masuk ke menu utamanya. Gambar 2.9 menunjukkan *library browser* dari Simulink.



Gambar 2.9. Simulink *library browser*

Dalam menu utama ini diperlihatkan *library* simulink yang terdiri dari kumpulan *blockset* dan *toolbox*. Pada *blockset* dan *toolbox* terdapat banyak blok yang memiliki fungsi-fungsi khusus. Tiap blok yang tersedia, dapat diketahui karakteristiknya dengan memasuki menu *help*. Untuk membuat suatu permodelan, maka kita harus membuat lembar kerja simulink yang baru (dapat diakses melalui *toolbar* yang tersedia). Lalu untuk menggunakan blok yang terdapat pada *library* kita cukup mengklik lalu tahan dan bawa blok yang dikehendaki menuju lembar kerja simulink. Gambar 2.10 menunjukkan lembar kerja dari simulink.



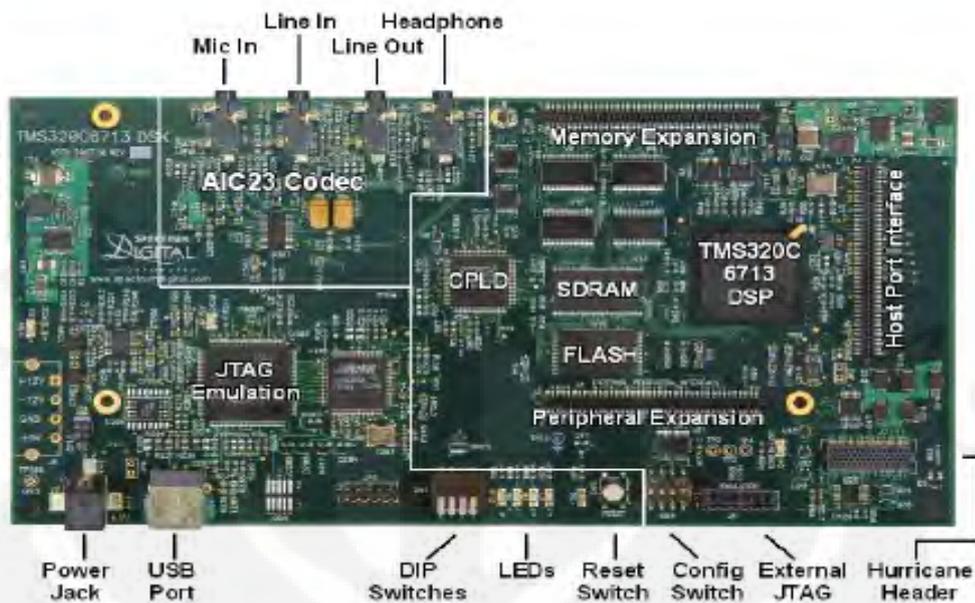
Gambar 2.10. Lembar kerja Simulink

Setelah kita selesai membuat suatu model, kita dapat menganalisisnya dengan mensimulasikan model tersebut, dengan cara memilih *icon –run–* pada *toolbar* yang tersedia. Pesan kesalahan akan muncul jika model tidak bisa disimulasikan. Saat mensimulasikan model, kita dapat memilih target simulasi, apakah hanya akan dijalankan pada komputer saja, atau dapat juga disimulasikan ke sebuah alat simulasi eksternal. Simulasi pada papan DSK merupakan contoh dari simulasi eksternal.

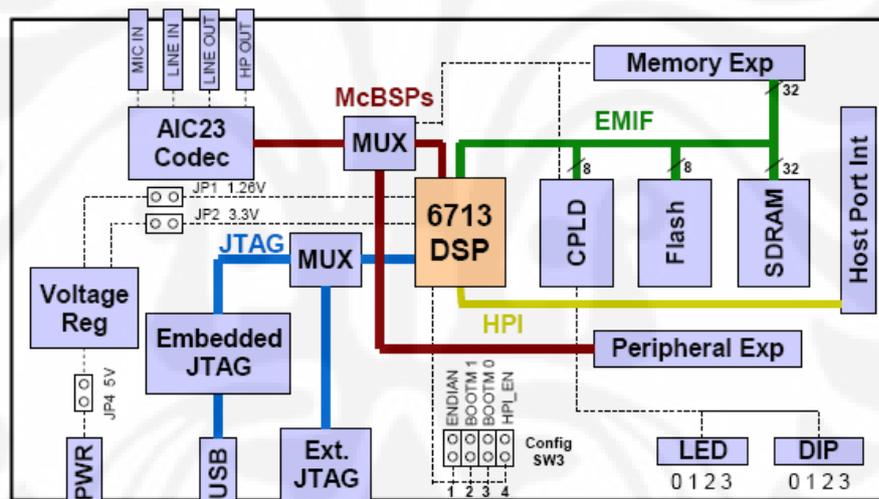
2.4 DSK TMS320C6713

DSK TMS320C6713 adalah salah satu DSP tipe C6000 yang dapat bekerja pada *fixed-point* maupun *floating-point*. Tetapi, DSP ini masih berupa *starter kit*, yaitu suatu *platform* yang dapat mensimulasikan DSP C6713 yang sebenarnya. DSK ini lebih ditujukan untuk keperluan edukasi, penelitian, serta evaluasi. Namun, hasil dari aplikasi yang kita buat di DSK ini sangat mungkin untuk diterapkan pada DSP C6713 yang sebenarnya.

Gambar 2.11 dan 2.12 menunjukkan gambaran fisik dan blok diagram dari DSK C6713.



Gambar 2.11 Bentuk fisik DSK TMS320C6713 [8]



Gambar 2.12 Blok diagram DSK TMS320C6713 [8]

Komponen-komponen utama dan pendukung dari DSK C6713 yaitu:

1. Prosesor TMS320C6713
Merupakan prosesor dengan kecepatan *clock* 225 MHz yang mendukung operasi *fixed-point* dan *floating-point*. Kecepatan operasinya dapat mencapai 1350 juta operasi *floating-point* per detik (MFLOPS) dan 1800 juta instruksi per detik (MIPS). Selain itu, prosesor ini dapat melakukan 450 juta operasi *multiply-accumulate* per detik.
2. CPLD (*Complex Programmable Logic Device*)

CPLD berisi register-register yang berfungsi untuk mengatur fitur-fitur yang ada pada board. Pada DSK C6713, terdapat 4 jenis register CPLD, yaitu:

- a. USER_REG Register
Mengatur *switch* dan LED sesuai yang diinginkan *user*.
- b. DC_REG Register
Memonitor dan mengontrol *daughter card*.
- c. VERSION Register
Indikasi yang berhubungan dengan versi *board* dan CPLD.
- d. MISC Register
Untuk mengatur fungsi lainnya pada *board*.

3. Flash memory

DSK menggunakan memori *flash* yang berfungsi untuk *booting*. Dalam *flash* ini berisi sebuah program kecil yang disebut POST (*PowerOn Self Test*). Program ini berjalan saat DSK pertama kali dinyalakan. Program POST akan memeriksa fungsi-fungsi dasar *board* seperti koneksi USB, *audio codec*, LED, *switches*, dan sebagainya.

4. SDRAM

Memori utama yang berfungsi sebagai tempat penyimpanan instruksi maupun data.

5. AIC23 Codec

Berfungsi sebagai ADC maupun DAC bagi sinyal yang masuk ke *board*.

6. Daughter card interface

Konektor-konektor tambahan yang berguna untuk mengembangkan aplikasi-aplikasi pada *board*. Terdapat 3 konektor, yaitu *memory expansion*, *peripheral expansion*, dan *Host Port Interface*.

7. LED dan Switches

LED dan *switches* ini merupakan fitur yang dapat membantu dalam membangun aplikasi karena dapat deprogram sesuai keinginan *user*.

8. JTAG (*Joint Test Action Group*)

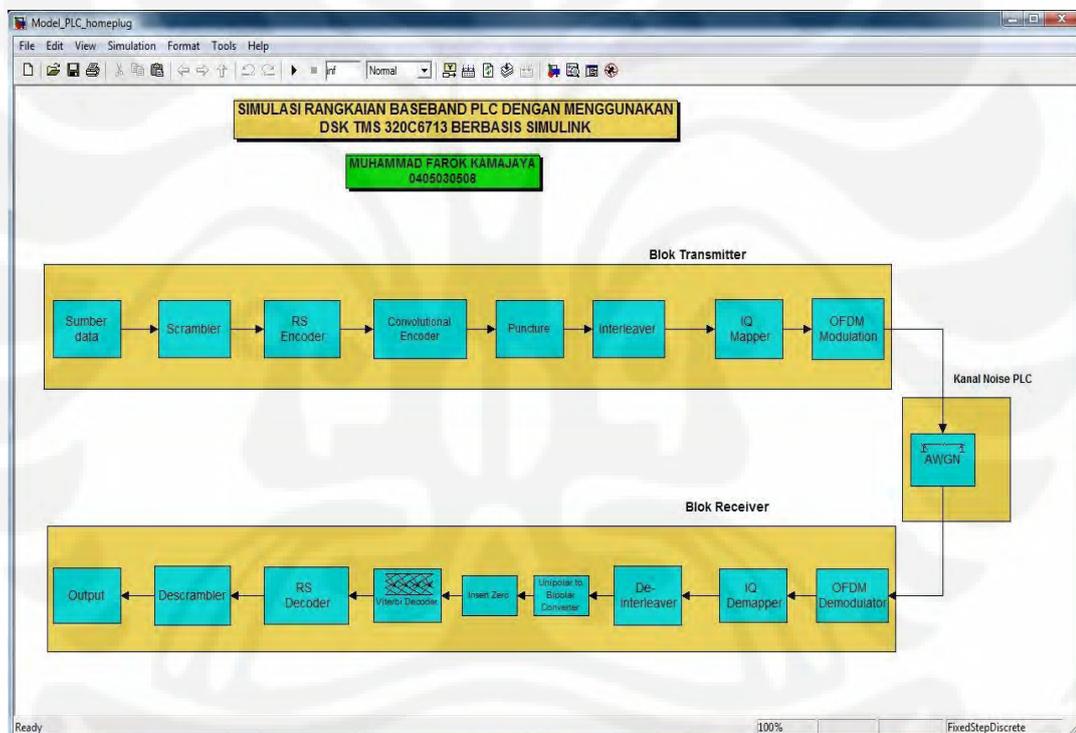
Merupakan konektor yang dapat melakukan transfer data dengan kecepatan yang sangat tinggi. Hal ini berguna dalam aplikasi *real-time*.

BAB III

PERANCANGAN SIMULASI RANGKAIAN BASEBAND PLC

3.1. Perancangan Model Baseband PLC dengan Simulink

Simulasi model rangkaian *baseband* PLC ini dibuat dengan menggunakan program Simulink yang terdapat pada MATLAB *versi* 7.0.4. Perancangan model rangkaian *baseband* PLC ini mengikuti standar yang ada pada *homeplug 1.0* sebagai dasar acuan utama dalam menentukan nilai parameter-parameter dari setiap blok diagramnya. Model umum dari rangkaian simulasi *baseband* PLC ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Model Simulasi rangkaian baseband PLC pada simulink

Sistem rangkaian model simulasi *baseband* PLC pada Gambar 3.1 terdiri dari 3 *sub-system*, yaitu: diagram blok transmitter, kanal noise PLC dan diagram blok *receiver*.

3.1.1 Diagram Blok Transmitter

3.1.1.1 Blok *Input data*

Blok ini terdiri dari 2 *sub-blok*, yaitu *sub-blok sample data* dan *sub-blok integer to bit converter*.

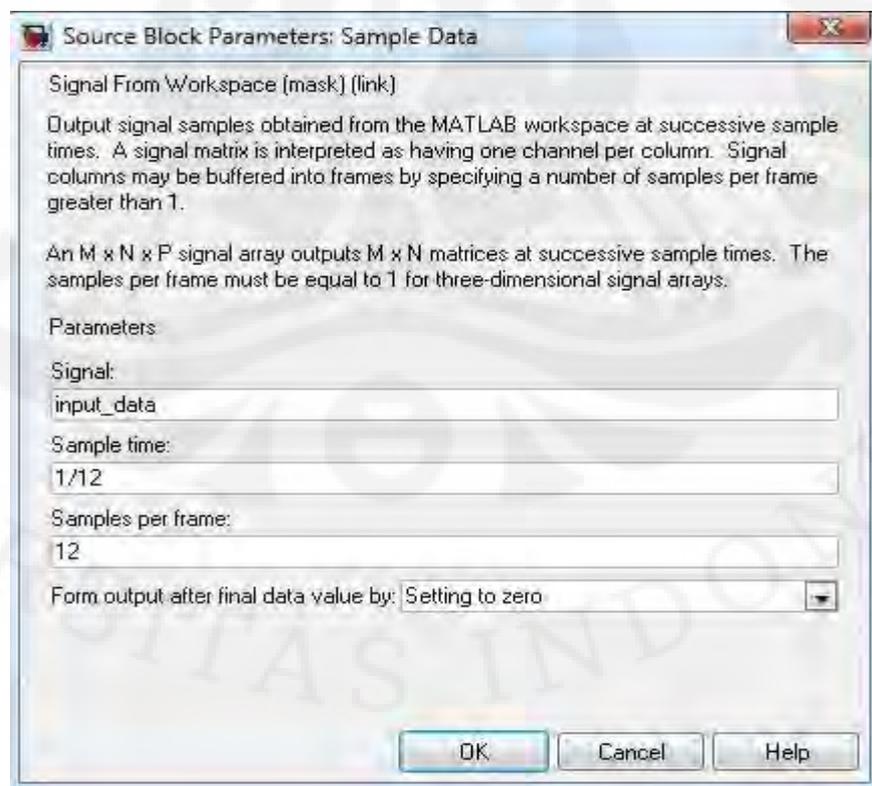
- *Sub-blok sample data*

Sub-blok ini berfungsi sebagai sumber data yang akan digunakan sebagai pesan informasi pada model simulasi *baseband* PLC ini. *Sub-blok* ini akan mengambil *input* data dari *workspace* sebagai sampel yang nilainya telah ditentukan sebelumnya dalam file ‘data.m’. Nilai *input* data pada file ‘data.m’ ini dinyatakan dalam baris vektor yang dibangkitkan dengan perintah:

```
input_data = sscanf(['32 50 28 4B 99 45 29 C4 79 A4 0F 55'], '%x');
```

Input data ini merupakan sampel data sebanyak 12 simbol dalam bilangan heksadesimal. Setelah data dimasukkan dalam *workspace* nilai data tersebut dikonversi menjadi bilangan desimal. Selanjutnya data dari *workspace* ini akan dibangkitkan dan diproses sebagai data masukan pada model simulasi *baseband* PLC yang telah dibuat.

Pada *sub-blok sample data* diperlukan penyesuaian parameter sesuai dengan input data yang dibangkitkan. Untuk parameter *sample time* yang digunakan adalah 1/12 artinya lama waktu yang digunakan untuk 1 *frame*-nya adalah 1 detik dengan keluaran sebanyak 12 simbol. Gambar 3.2 dibawah menunjukkan parameter dari blok *sample data*.



Gambar 3.2 Parameter Blok *Sample Data*

- Integer to Bit converter

Sub-blok ini berfungsi untuk mengubah setiap nilai integer (bilangan bulat) yang berasal dari input data menjadi data digital dalam bit-bit. Pada blok *input* data, data yang dikirim berupa paket data (*frame*) menjadi bilangan biner (*bit stream*) dengan lebar 8 bit per integer. Nilai dari bit ini nantinya akan di XOR-kan dengan nilai bit yang berasal dari *PN sequence generator*.

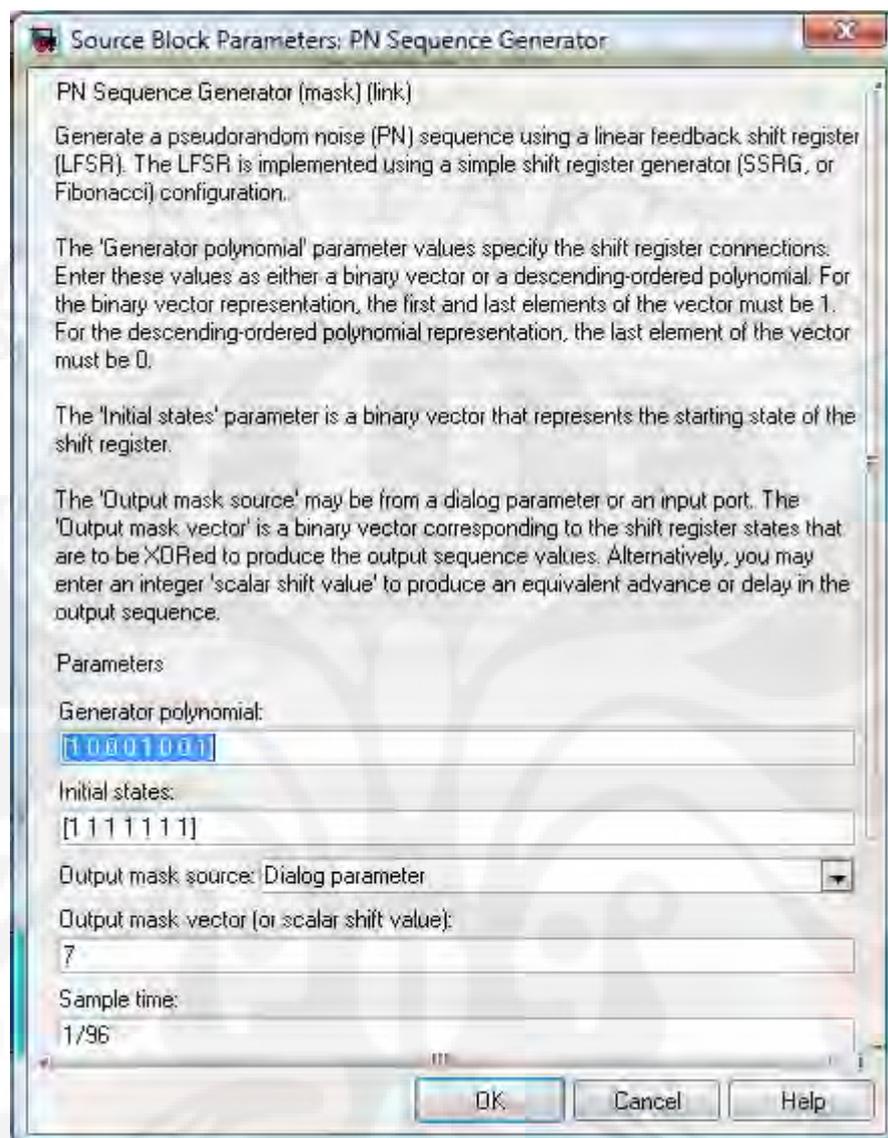
3.1.1.2 Blok Scrambler

Blok *scrambler* berfungsi sebagai pengacakan data untuk meningkatkan keamanan dari sebuah pembawa, menjamin jumlah yang cukup dari bit transmisi dan membantu *clock recovery*. Blok ini terdiri dari 3 *sub-blok*, yaitu: *sub-blok* *PN Sequence Generator*, *sub-blok* XOR dan *sub-blok* zero pad.

- *Sub-blok* *PN Sequence Generator*

Sub-blok *PN Sequence Generator* berfungsi untuk membangkitkan suatu nilai *pseudorandom binary*. Aliran bit data (*bit stream*) yang keluar dari blok *input* data selanjutnya akan diacak dengan bit-bit keluaran yang berasal dari *PN Sequence Generator*. Metode pengacakan yang digunakan adalah dengan menggunakan *polynomial sequence* $p(x) = x^7 + x^4 + 1$ dengan 7 bit yang diprogram. Bentuk *polynomial* ini kemudian diubah ke dalam bentuk vektor menjadi [1 0 0 0 1 0 0 1]. Nilai vektor inilah yang menjadi parameter *generator polynomial* dari *sub-blok* *PN sequence generator*.

Pada setiap pengiriman *frame* data, isi dari *register pseudorandom binary sequence* dibersihkan dan diisi kembali dengan deretan bit [1 1 1 1 1 1 1] sebagai nilai *initial states*. Sedangkan parameter *sample time* yang digunakan adalah 1/96 dengan keluaran yang dihasilkan adalah dalam bentuk *frame* dimana *samples per frame*-nya bernilai 96 bit. Nilai bit ini sama dengan keluaran dari blok *input* data sehingga proses pengacakan dapat dilakukan. Gambar 3.3 menunjukkan parameter dari blok *PN Sequence generator*.



Gambar 3.3 Parameter Blok PN Sequence Generator

- *Sub-blok XOR*

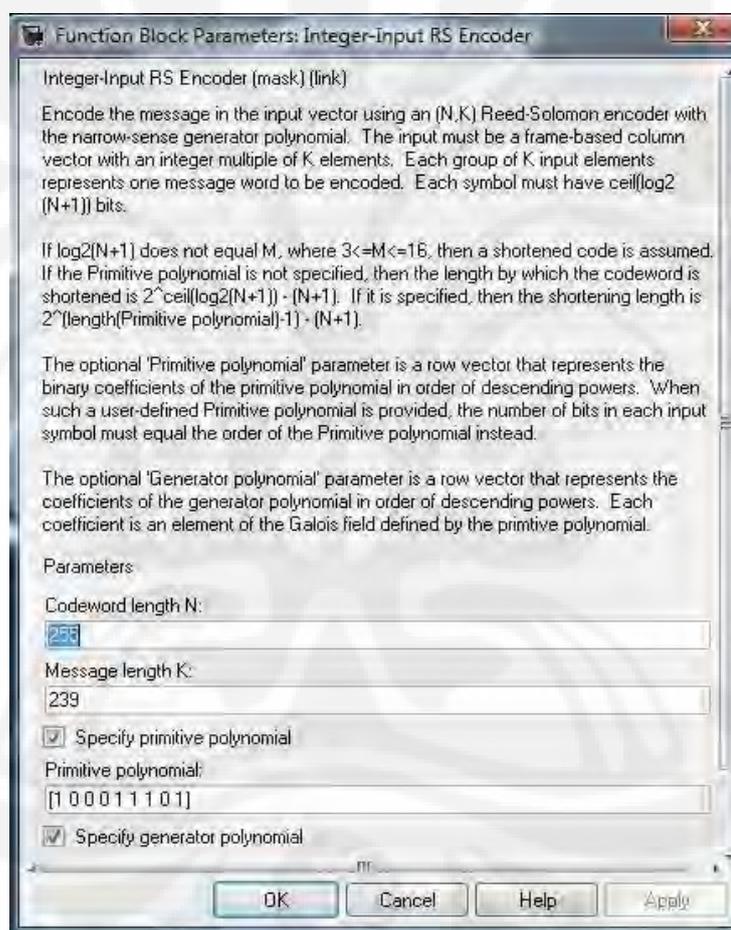
Blok *PN sequence generator* ini akan membangkitkan serangkaian *bit* yang unik. *PN Code* merupakan serangkaian *bit* ini memiliki pola perulangan yang unik. Setiap perulangan ini kemudian akan di-*XOR* dengan *bit input*.

Tabel 3.1 Tabel XOR

bit 1	bit 2	hasil XOR
0	0	0
0	1	1
1	0	1
1	1	0

3.1.1.3 Blok Reed Solomon Encoder

Pengkodean Reed Solomon digunakan pada pangkodean terluar untuk mengatasi *burst error*. Masukan pengkodean Reed-Solomon yang digunakan pada model memiliki 256 level sinyal yang lebih dikenal dengan istilah *bytes*. Panjang data masukan pengkodean luar (k) adalah sepanjang 239 bytes, panjang keluaran *codeword*-nya (n) adalah sepanjang 255 bytes. Maka nilai m , yakni besar Galois Field (GF) yang menentukan nilai *primitive polynomial* $p(x)$ sebagai nilai koreksinya adalah sebesar $(n-k)/2$ yakni 8 byte. Gambar 3.4 memperlihatkan parameter pada blok *reed solomon*.

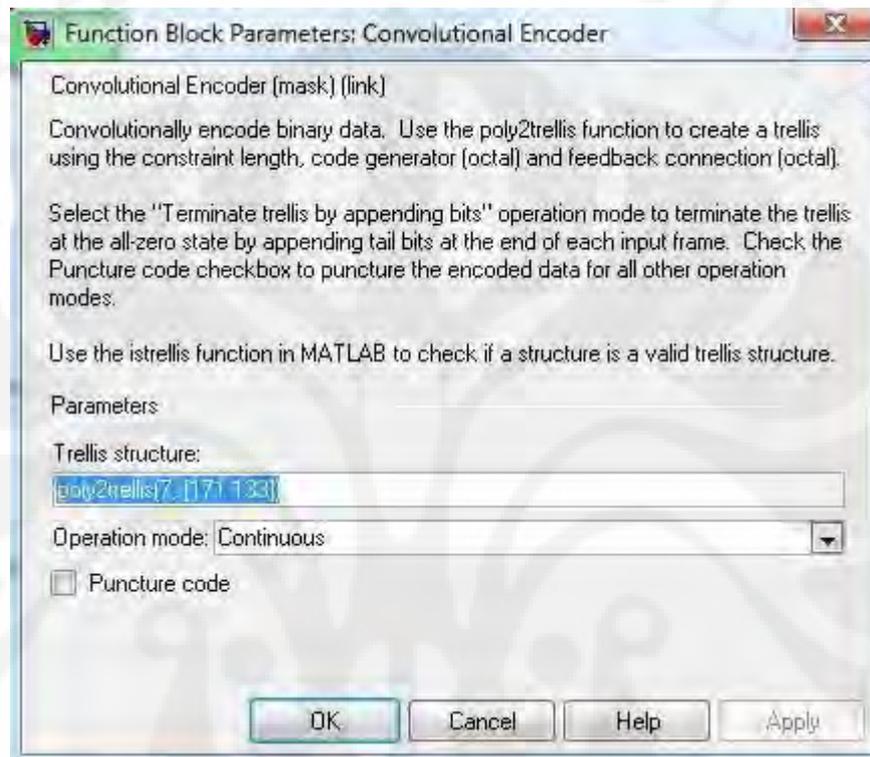


Gambar 3.4 Parameter Blok Reed Solomon

3.1.1.4 Blok "Convolutional Encoder"

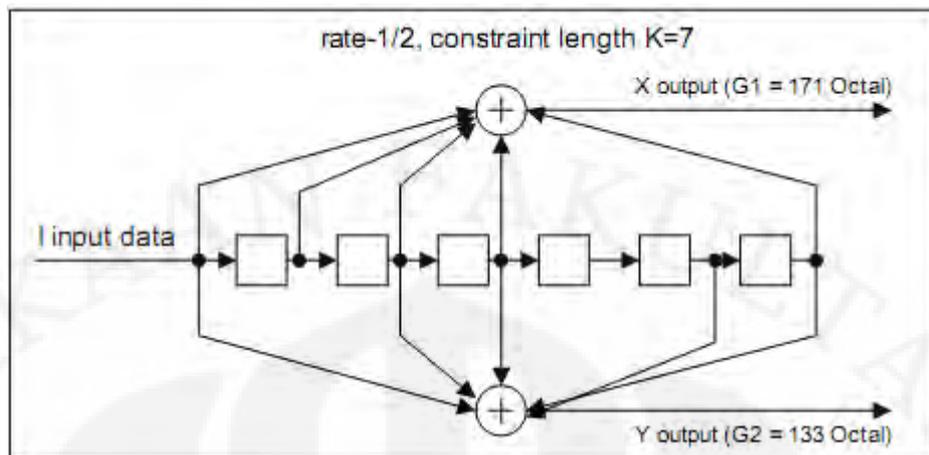
Pada simulasi ini digunakan *convolutional code* dengan *code rate* $\frac{1}{2}$ dan *constraint length* $K = 7$. Hal ini menunjukkan bahwa *encoder* memiliki satu buah *input* dan menghasilkan dua buah *output*. *Generator Polynomial* yang digunakan adalah $\{\text{poly2trellis}(7, [171 \ 133])\}$. Bilangan 171 133 merupakan bilangan *octal*

yang menunjukkan keadaan awal dari isi register seperti ditunjukkan Gambar 3.5. Bilangan tersebut menunjukkan bilangan biner 1111001 dan 1011011. Bilangan biner tersebut apabila dibaca menurut aturan bilangan octal maka akan menjadi 1 111 001 dan 1 011 011. Gambar 3.5 menunjukkan parameter blok *convolutional encoder*.



Gambar 3.5 Parameter Blok *Convolutional Encoder*

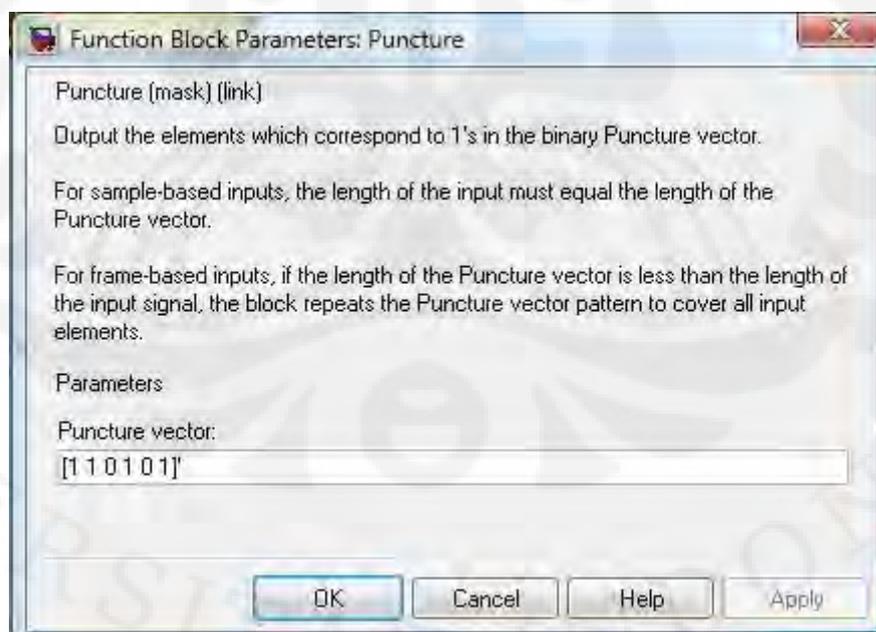
Pada *convolutional encoding*, *bit* yang dihasilkan oleh *encoder* merupakan hasil sistem *xor* dari beberapa *bit* masukan sebelumnya dengan *bit input* pada saat itu. Sistem *xor* tersebut dapat dirancang sendiri. Sistem *xor* ini mencakup bagaimana alur operasi *xor* dan berapa lama suatu *bit* akan mempengaruhi keluaran *bit* lain. Keuntungan *convolutional encoder* adalah dapat didesain untuk mampu mendeteksi dan terutama memperbaiki *error* dengan lebih baik dibandingkan dengan *Linear Block Encoding*. *Convolutional encoder* dapat dirancang dengan menggunakan beberapa *shift register* sederhana. Gambar 3.6 menunjukkan bagan dari *convolutional encoder*.



Gambar 3.6 Bagan ilustrasi rangkaian *convolutional encoder* $\frac{1}{2}[3]$

3.1.1.5 Blok Puncture

Blok *puncture* berfungsi untuk mempercepat *coding rate* dari bit data yang masuk dan mengurangi jumlah bit yang ditransmisikan. Pada simulasi ini vektor *puncture* yang digunakan adalah [1 1 0 1 0 1] yang akan menghasilkan *output rate* sebesar $\frac{3}{4}$. Artinya perbandingan antara bit keluaran dan bit masukan pada blok *puncture* sebesar $\frac{3}{4}$, atau jumlah bit yang keluar berkurang $\frac{1}{4}$ dari total bit masukan. Berkurangnya jumlah bit yang ditransmisikan inilah yang mempercepat *coding rate* dari bit data. Gambar 3.7 menunjukkan parameter dari blok *puncture*.



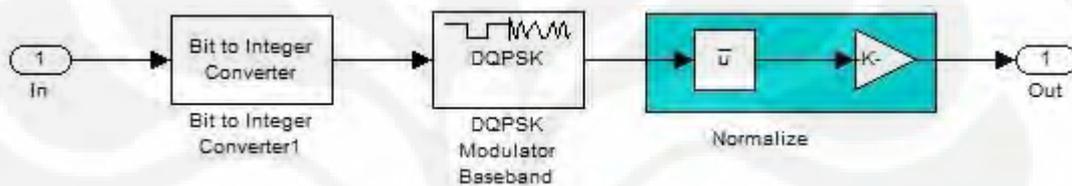
Gambar 3.7 Parameter Blok *Puncture*

3.1.1.6 Blok *Interleaver*

Pada blok *interleaver*, bit-bit *output* dari blok *puncture* kemudian akan diatur ulang berdasarkan konsep permutasi acak. Pengaturan ini untuk memisahkan bit-bit yang berdekatan. Banyaknya elemen yang diacak sama dengan jumlah sampel yang masuk kedalam blok ini, yaitu sebanyak 12 sampel. *Interleaver* digunakan untuk meningkatkan kemampuan *error correction* dari skema *coding* yang telah digunakan.

3.1.1.7 Blok *IQ Mapper*

Blok *IQ mapper* terdiri dari beberapa sub-blok yang ditunjukkan pada Gambar 3.8



Gambar 3.8 Blok diagram pada IQ Mapper

- *Sub-blok Bit to Integer Converter*

Blok ini bertujuan untuk mengubah data bit ke dalam bilangan bulat (*integer*).

- *Sub-blok DQPSK Modulator Baseband*

Blok ini merupakan modulasi *baseband* DQPSK yang menghasilkan sinyal keluaran termodulasi fasa.

- *Sub-blok Normalize*

Konstelasi yang dihasilkan pada model ini merupakan konstelasi yang belum dinormalisasikan. Untuk melakukan normalisasi sehingga didapatkan daya rata-rata sama dengan satu, maka setiap konstelasi harus dikalikan dengan faktor normalisasi. Untuk modulasi DQPSK faktor normalisainya adalah $1/\sqrt{2}$.

3.1.1.8 Blok “OFDM Simbol”

Blok OFDM modulation ini adalah sebuah subsystem yang berfungsi untuk memodulasikan sinyal informasi yang kita miliki dari blok sebelumnya. Blok ini terdiri dari beberapa blok dasar sebagai berikut:

- *Sub-blok Multiport Selector*

Dalam blok *Multiport Selector* ini mode yang digunakan adalah *select row*. *Select row* ini digunakan untuk mengubah data serial menjadi paralel.

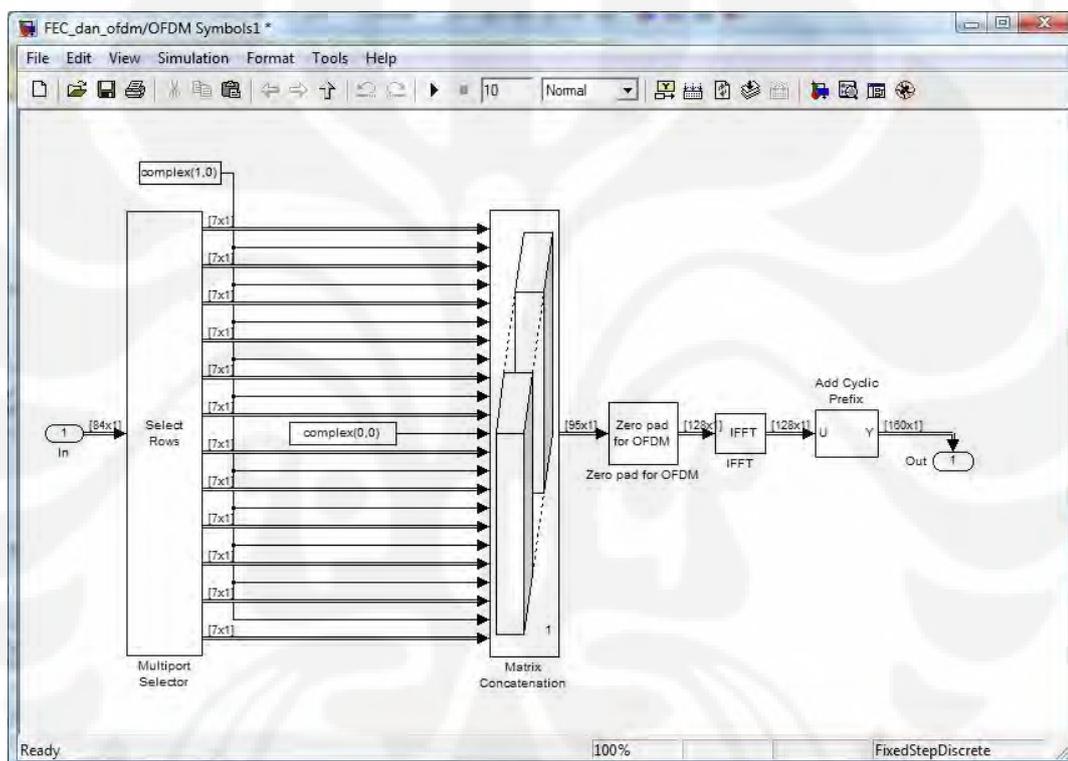
- Sub-Blok IFFT

Blok ini berfungsi untuk melakukan transformasi dari domain frekuensi ke domain waktu.

- Sub-Blok *Add Cyclic Prefix*

Penggunaan *add cyclic prefix* adalah untuk mengkompensasi ketidak-sinkronan dari sinyal yang akan diterima. Blok ini mengerjakan bagian penambahan cyclic prefix pada sinyal OFDM yang telah terbentuk dari IFFT.

Blok OFDM modulation ditunjukkan oleh Gambar 3.9



Gambar 3.9 Blok OFDM Modulation

Pada model PLC yang digunakan 84 data *subcarrier*. Hal ini sesuai dengan parameter dari *homeplug 1.0* seperti yang terdapat pada Tabel 3.2

Tabel 3.2 Parameter PLC *homeplug 1.0*

Number of Subcarriers	84
Subcarrier Spacing	195.3125 kHz
Symbol Period	8.4 usec
FFT Interval	5.12 usec
Subcarrier Modulation	BPSK (Frame Control) DBPSK (Data, ROBO) DQPSK (Data)
Peak PHY Rate	14 Mbps
Minimum PHY Rate (ROBO)	1 Mbps

Pada simulasi digunakan modulasi DQPSK yang memetakan sebanyak 2 *bit per symbol*. Maka akan didapatkan $\text{bits/OFDM symbol} = 2 \times \text{jumlah data subcarrier} = 2 \times 84 = 168$ bits. Pada model digunakan *code rate puncture* sebesar $\frac{3}{4}$ pada blok *convolutional code*, maka nilai untuk $\text{bits/OFDM symbol} = \frac{3}{4} \times 168 = 126$ bits. Pada blok *Reed Solomon* memiliki *code rate* sebesar $\frac{239}{255}$ sehingga $\text{bits/OFDM symbol} = \frac{239}{255} \times 126$ bits = 118,094 bits. Dengan periode simbol sebesar 8,4 μs , maka besarnya $\text{bit rate} = \frac{118,094 \text{ bits}}{8,4 \mu\text{s}} = 14,058$ Mbps. Hal ini sesuai dengan parameter pada Tabel 3.1 yang menunjukkan nilai *bit rate* sebesar 14 Mbps.

3.1.2 Kanal *noise* PLC

Blok AWGN ini merepresentasikan blok kanal transmisi pada PLC yang berfungsi sebagai gangguan pada saluran transmisi. Gangguan AWGN ini dinyatakan dengan parameter *Signal to Noise Ratio* (SNR). Pada simulasi ini digunakan nilai SNR 30dB dengan daya 0,01 W.

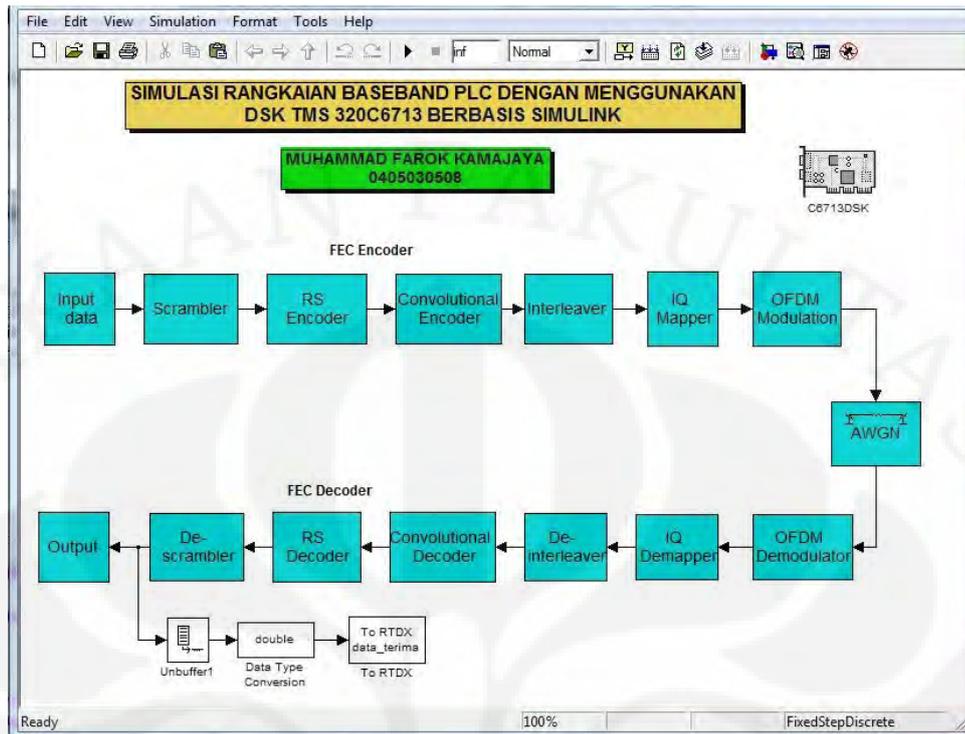
3.1.3 Blok *Receiver*

Proses yang terjadi pada sisi *receiver* merupakan proses pembalikan pada *transmitter*.

3.2 Penerapan Model Simulink pada DSK TMS320C6713

Pada MATLAB disediakan suatu alat bantu yaitu Target Support Package™ dimana memungkinkan pengguna untuk menggunakan *Real-Time® Workshop software* untuk menghasilkan program dengan *platform* bahasa C dimana merupakan hasil dari implementasi dari model yang telah dibuat di Simulink®. Dari hasil yang berupa bahasa C, pengguna dapat meng-*compile*, melakukan koneksi, dan menjalankan dengan *hardware* C6713 DSP *Starter Kit* (DSK).

Pada Gambar 3.10 menunjukkan blok diagram system secara keseluruhan. Sistem tersebut akan diaplikasikan dengan DSK TMS320C6713.



Gambar 3.10 Model Simulasi dengan DSP Board

Berikut ini adalah tambahan blok yang dipakai dalam model simulasi yang terdapat pada Gambar 3.10.

- Blok C6713 DSK

Blok ini berfungsi untuk melakukan *targetting* simulasi dengan alat DSK.

- Blok *Unbuffer*

Blok ini berfungsi untuk mengubah *frame based* menjadi *sampel based* dengan data rate yang lebih tinggi.

- Blok To RTDX

Blok ini berfungsi untuk mengambil data dari DSK yang terhubung ke blok.

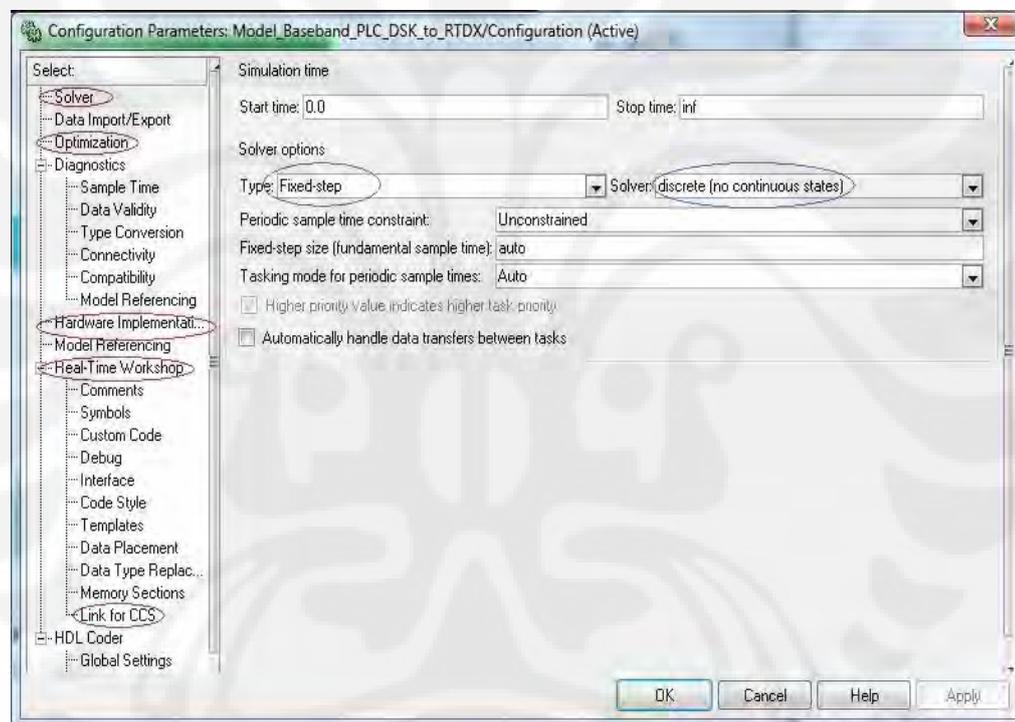
Setelah model yang kita buat sudah selesai, ada beberapa parameter konfigurasi yang harus diubah. Hal ini dapat kita lakukan dengan memilih ***Simulation*** → ***Configuration Parameters***. Ada beberapa yang diubah yaitu:

1. Pada *tab solver*, mengubah *type* menjadi *fixed-step* dan *solver* menjadi *discrete*.
2. Pada *tab optimization*, melakukan *uncheck block reduction* dan *implement logic signal as boolean data* yang terdapat pada menu *simulation and code generation*.

3. Pada tab *hardware implementation*, mengubah *device type* yang terdapat pada menu *embed hardware* menjadi TI C6000.
4. Pada tab *real time workshop* di menu *target selection*, mengubah *system target file* menjadi *ccslink_ert.tlc*.
5. Pada tab *real time workshop* subtab *debug*, melakukan *check verbose build* pada menu *build process*.
6. Pada tab *real time workshop* subtab *Link for CCS* di menu *project option*, mengubah *system stack size* menjadi 8192.

Pengaturan konfigurasi parameter pada model simulink dapat dilihat pada

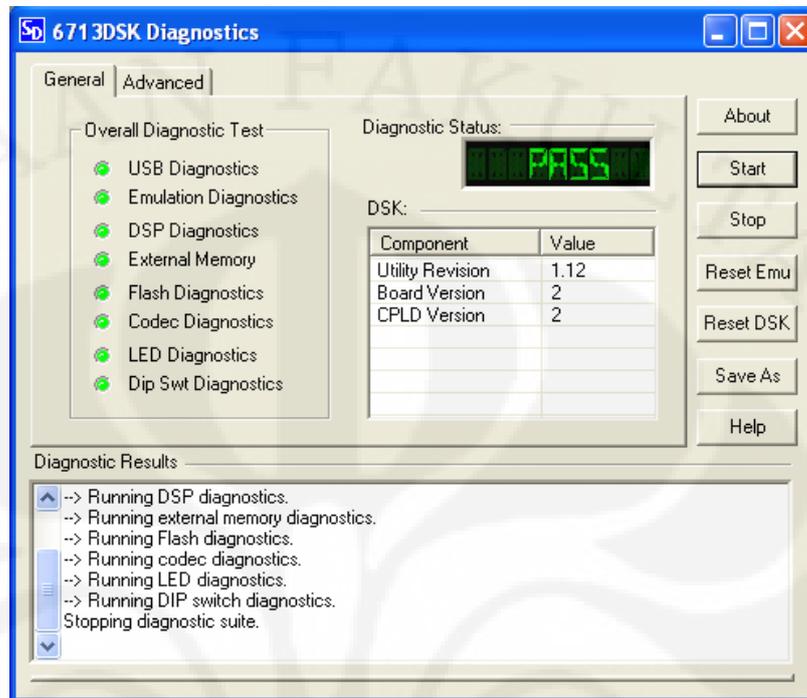
Gambar 3.11 dibawah ini:



Gambar 3.11 Konfigurasi parameter Model

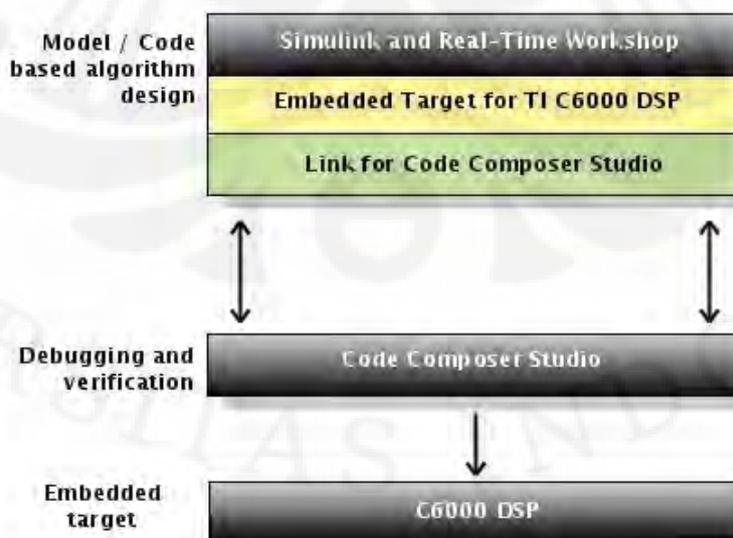
Setelah model rangkaian *baseband* PLC telah disesuaikan dan data input telah tersimpan pada *workspace* Matlab. Selanjutnya dilakukan pemeriksaan koneksi perangkat DSK ke komputer. DSK harus terhubung ke komputer menggunakan kabel USB dan untuk memastikan keduanya terhubung dengan benar maka proses diagnosa perlu dilakukan. Caranya adalah dengan menggunakan program *6713 diagnostic* yang disediakan oleh CCS. Jika DSK terhubung dengan baik ke komputer, maka di kotak dialog program *diagnostic* tersebut akan muncul pesan 'PASS'.

Gambar 3.12 menunjukkan proses *diagnostic* yang berhasil menghubungkan perangkat DSK dengan komputer.



Gambar 3.12 Proses *Diagnostic*

Secara sederhana proses dalam *targetting* ini menggunakan simulink dan CCS. Untuk menghubungkan simulink dengan DSK dibutuhkan *Real Time Workshop*, *Embedded Target for TI C6000 DSP*, dan *Link for CCS*. Ketiga hal tersebut dapat ditemukan di Simulink dan harus dilakukan pengaturan konfigurasi. Hubungan ketiga hal tersebut dapat dilihat pada Gambar 3.13.



Gambar 3.13 Diagram Alir *Targetting* to C6000 DSP[10]

Gambar 3.13 menunjukkan proses *debugging* dan *verification* dilakukan oleh *software* CCS. Penggunaan CCS memungkinkan untuk menghasilkan kode-kode yang akan digunakan dalam C6000 DSP sehingga tidak diperlukan lagi pembuatan program dengan manual karena sudah dilakukan oleh CCS.

Setelah pengaturan dan konfigurasi dilakukan dengan benar maka dilakukanlah proses *incremental build* pada model. Selanjutnya Matlab akan membuat kode program dalam bahasa C seperti pada Gambar 3.14 di bawah ini.



```

Command Window
To get started, select MATLAB Help or Demos from the Help menu.

-alinkerOptionsStr=""
-aoverrunNotificationMethod="None"
-aoverrunNotificationFcn="myfunction"
-aconfigurePIL=0
-aconfigPILBlockAction="None"
-astackSize=8192
-p10000

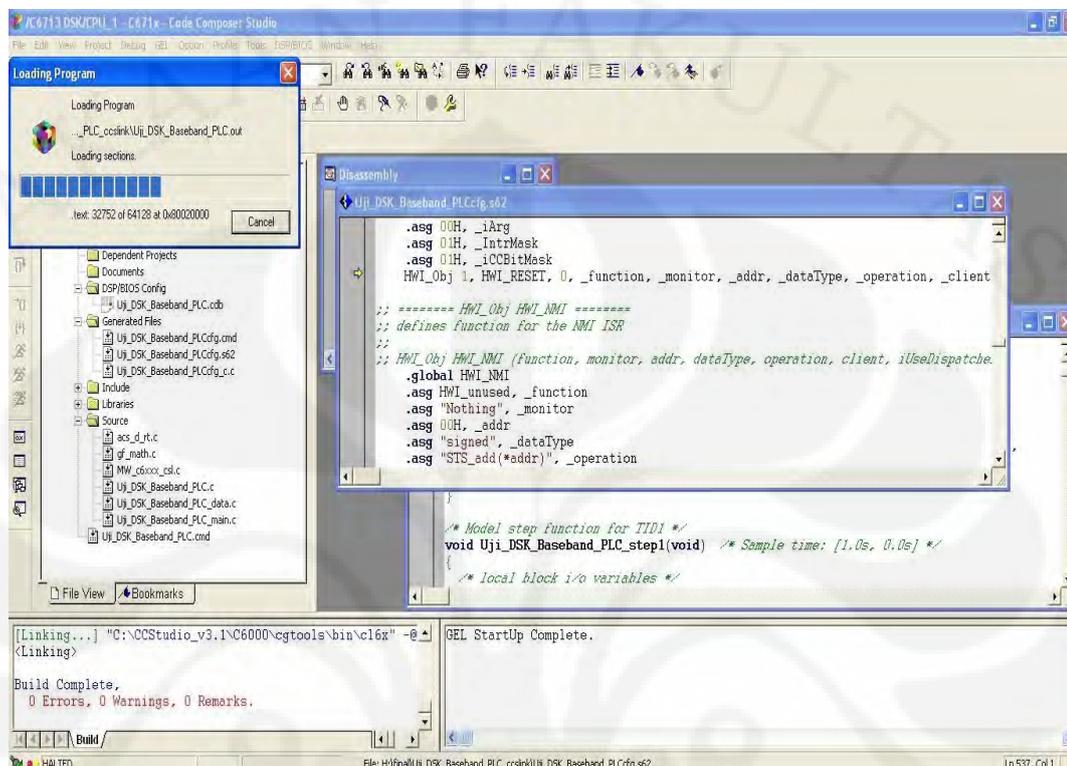
### Loading TLC function libraries
.....
### Initial pass through model to cache user defined code
.....
### Caching model source code
.....
### Writing header file Uji_DSK_Baseband_PLC_types.h
.
### Writing header file Uji_DSK_Baseband_PLC.h
### Writing source file Uji_DSK_Baseband_PLC.c
### Writing header file Uji_DSK_Baseband_PLC_private.h
.
### Writing source file Uji_DSK_Baseband_PLC_data.c
### Writing source file Uji_DSK_Baseband_PLC_main.c
### TLC code generation complete.
.### Generating the DSP/BIOS configuration file ...
### Creating project in Code Composer Studio(tm)...
### Building Code Composer Studio(tm) project...
### Build complete
### Downloading COFF file
### Downloaded: Uji_DSK_Baseband_PLC.out
>>

```

Gambar 3.14 Proses pembuatan program C

Incremental Build  yang dilakukan oleh Matlab akan memanggil program CCS 3.1 yang sebelumnya sudah terinisialisasi di dalam komputer. Proses selanjutnya dilakukan oleh *software* CCS 3.1 untuk menghasilkan kode dengan bahasa C. Gambar 3.14 menunjukkan proses *build* yang selanjutnya akan ditangani oleh CCS dimana akan menghasilkan *file* dengan ekstensi out yang

merupakan hasil dari rancangan yang dibuat di Matlab. *File* inilah yang akan digunakan untuk diaplikasikan ke *hardware*. Gambar 3.15 menunjukkan proses *load program* yang dilakukan oleh software CCS.



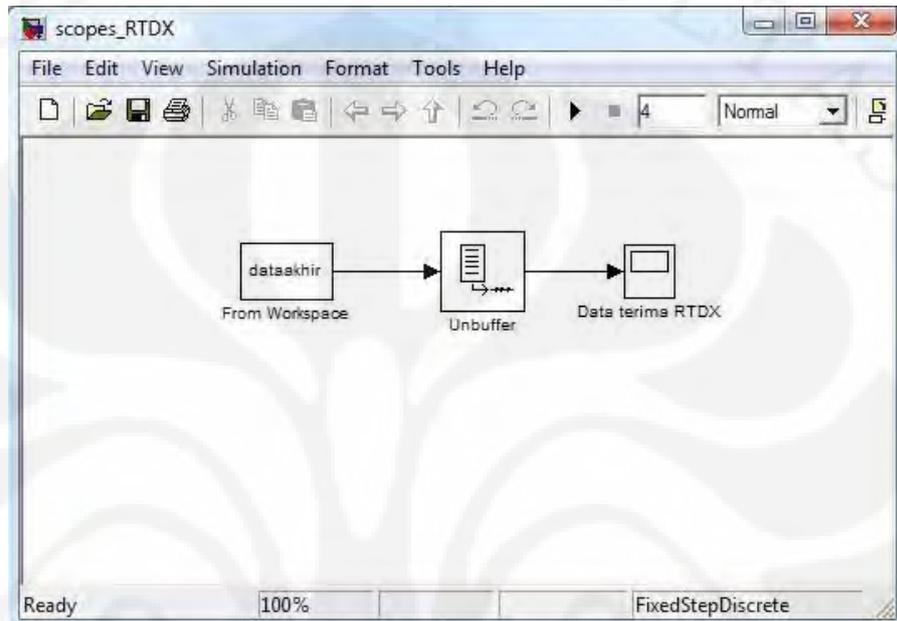
Gambar 3.15 Proses *Load Program*

Untuk memperoleh data keluaran dari model simulink yang dijalankan pada DSK diperlukan program *driver* untuk RTDX yang tersimpan dalam format M-file (terlampir). Program tersebut berupa perintah-perintah yang berfungsi untuk menjalankan program yang sudah terpasang di DSK, membuka jalur RTDX, melakukan pertukaran data dari DSK ke Matlab dan menghentikan jalannya program. Dalam program *driver* juga ditentukan lama proses berjalannya program di DSK dan pertukaran datanya dengan Matlab. Untuk menggunakan file tersebut dilakukan dengan cara mengetik nama RTDXdriver('path/filename') pada *command window*. Pada file tersebut terdapat fungsi RTDXdriver yang akan melakukan serangkaian prosedur untuk membaca data dari RTDX output channel. Bentuk program *driver* RTDX dapat dilihat pada lampiran.

Setelah program dalam DSK selesai bekerja, data yang telah dimuat dari DSK melalui RTDX disimpan pada *directory* yang telah ditentukan dengan format .mat dengan nama variabel sesuai yang telah dibuat pada program *driver*.

Untuk membaca dan memplot data yang telah diperoleh ke dalam grafik, diperlukan model baru pada Simulink. Sebelum model dijalankan, data *output* dalam format .mat harus berada di *workspace* Matlab.

Gambar 3.16 menunjukkan model Simulink untuk menampilkan data *output* pada pengujian RTDX.



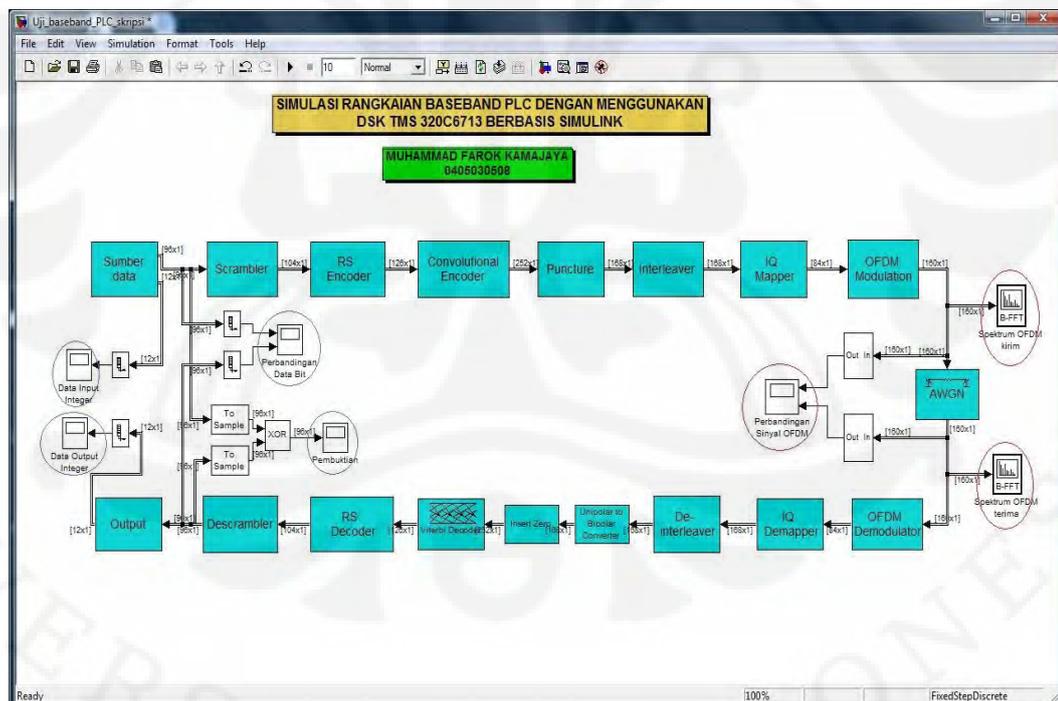
Gambar 3.16 Model untuk plot data keluaran RTDX

BAB IV UJICOBA DAN ANALISIS HASIL SIMULASI

Pengujian terhadap model rangkaian *baseband* PLC dilakukan dengan dua metode, yaitu pengujian simulasi menggunakan simulink dan pengujian dengan perangkat DSK TMS320C6713. Kedua metode pengujian tersebut dilakukan dengan cara membandingkan data yang telah diproses di *receiver* dengan data yang dikirim dari *transmitter*.

4.1 Hasil Pengujian dengan Simulink

Untuk melakukan pengujian simulink, model simulasi *baseband* PLC yang telah dibuat perlu ditambahkan blok *scope* untuk melihat bentuk sinyal yang diproses pada sistem rangkaian *baseband* PLC. Dari blok *scope* inilah data kemudian ditampilkan dalam grafik untuk membandingkan bentuk sinyal dan mempermudah dalam analisa. Gambar 4.1 adalah gambar rangkaian *baseband* PLC setelah ditambahkan blok *scope*.



Gambar 4.1 Pencantuman blok *scope* pada model simulink

Pada uji Simulink akan dilakukan perbandingan data kirim dan data terima, sinyal OFDM kirim dan OFDM terima pada sistem rangkaian *baseband* PLC. Data yang akan diambil dalam pengujian simulink ini adalah bentuk sinyal pada

sisi *transmitter* dan sisi *receiver*. Data kirim dan data terima yang akan dibandingkan berupa data dalam bentuk integer maupun dalam bit-bit. *Scope* pembuktian digunakan untuk membuktikan kebenaran data yang dikirim dengan data yang diterima melalui bantuan operasi *logical XOR*. Apabila grafik pembuktian XOR menunjukkan garis lurus maka data yang diterima sama dengan data kirim. Sedangkan bentuk sinyal OFDM yang dibandingkan adalah sinyal OFDM dalam domain waktu dan spektrum OFDM dalam domain frekuensi.

4.1.1 Hasil Data *Input* dan Data *Output*

Pada pengujian simulink, *input* data merupakan data *from workspace* yang telah ditentukan sebelumnya. Berikut adalah data *input* yang digunakan sebagai sumber data pada model simulasi *baseband PLC*:

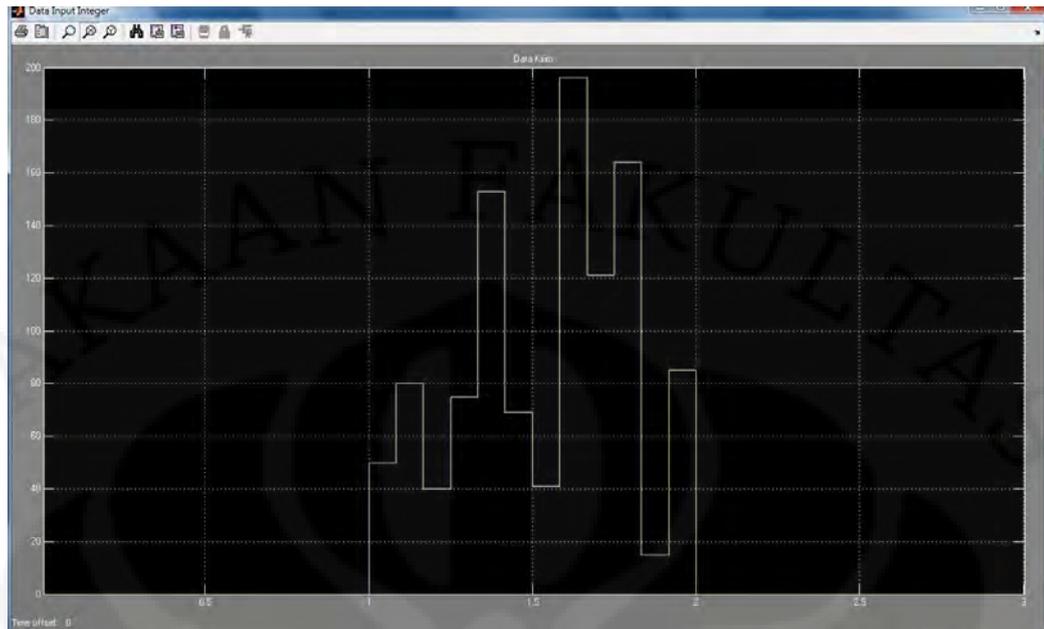
```
input_data = sscanf(['32 50 28 4B 99 45 29 C4 79 A4 0F 55'], '%x');
```

Data tersebut digunakan sebagai data masukan pada sistem dimana dalam 1 *frame*-nya terdapat 12 simbol yang digunakan sebagai sampel data. Tabel 4.1 menunjukkan nilai data *input* yang dibangkitkan beserta nilai bit-bit yang bersesuaian.

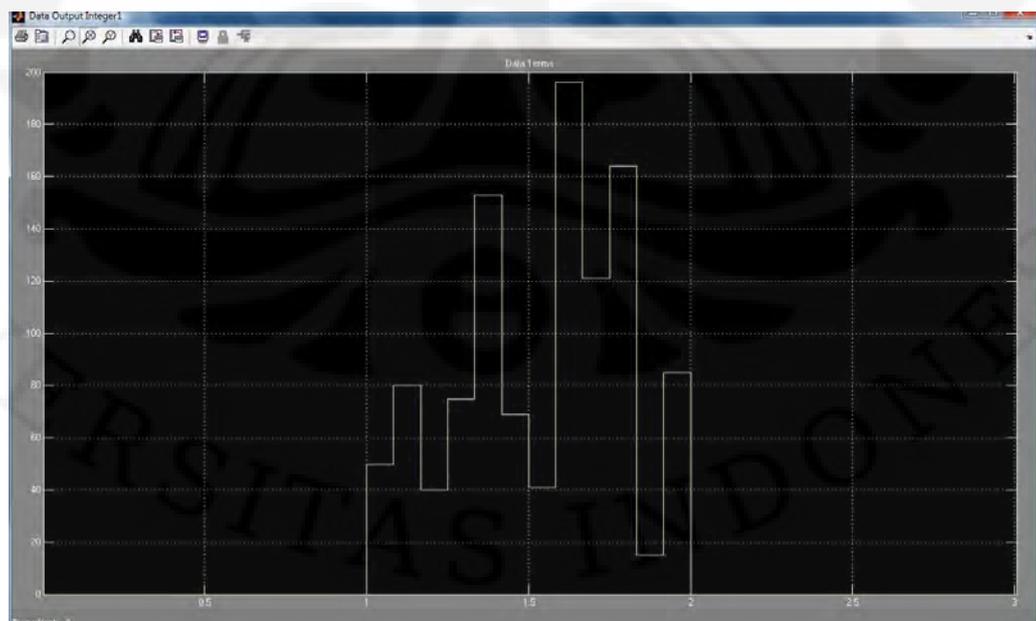
Tabel 4.1 Variabel “data *input*”

Elemen	sampel (hexa)	integer	Bit
1	32	50	00110010
2	50	80	01010000
3	28	40	00101000
4	4B	75	01001011
5	99	153	10011001
6	45	69	01000101
7	29	41	00101001
8	C4	196	11000100
9	79	121	01111001
10	A4	164	10100100
11	0F	15	00001111
12	55	85	01010101

Input data yang berupa sampel data ini akan dibangkitkan oleh Simulink dalam bentuk data *integer*. Nilai *input* data integer yang menjadi sumber data ini dapat dilihat pada Gambar 4.2.

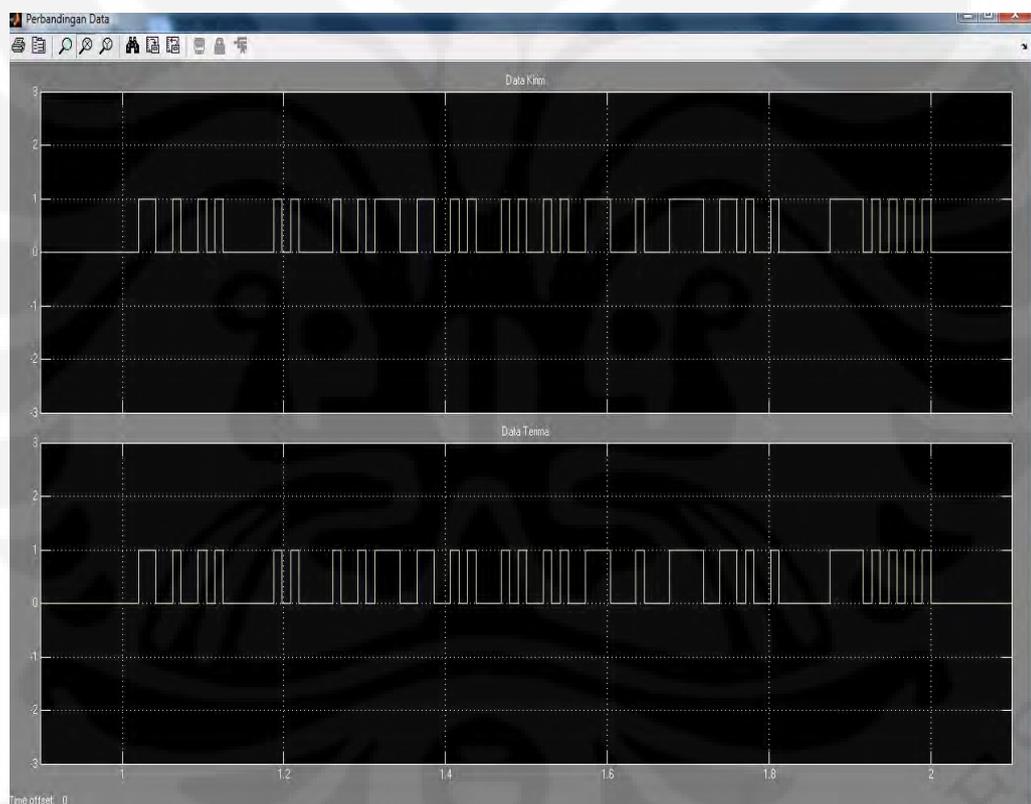
Gambar 4.2 Data *input integer*

Setelah data *input* melalui proses pada sisi transmisi (*encoding* dan modulasi) kemudian melewati kanal *noise* serta mengalami proses pembalikan (*decoding* dan demodulasi) pada sisi receiver maka diperoleh data *output* yang berada pada blok *output*. Untuk melihat bentuk sinyal data *output* setelah melalui sistem pemodelan dilakukan plot grafik seperti pada data *input*. Bentuk sinyal dari data *output* integer dapat dilihat pada Gambar 4.3 yang merupakan pesan akhir dari proses pengiriman data.

Gambar 4.3 Data *Output Integer*

Hasil data *output* pada Gambar 4.3 memiliki nilai yang sama seperti data *input* pada Gambar 4.2. Hal ini membuktikan bahwa data yang dikirim telah berhasil diterima dengan baik.

Pada blok *data input* terdapat *sub-blok integer to bit converter* yang akan mengubah data integer menjadi data bit dengan nilai seperti yang terdapat pada tabel 4.1. Perubahan menjadi data bit ini karena proses *transmisi* berlangsung secara digital berupa *bit stream* yang bernilai “1” dan “0”. Nilai dari data kirim ini kemudian diplot pada grafik untuk melihat bentuk sinyal data kirim yang dibangkitkan. Bentuk sinyal dari data kirim berupa *bit stream* dapat dilihat pada Gambar 4.4 pada bagian atas, sedangkan data terima dapat dilihat pada Gambar 4.4 bagian bawah.



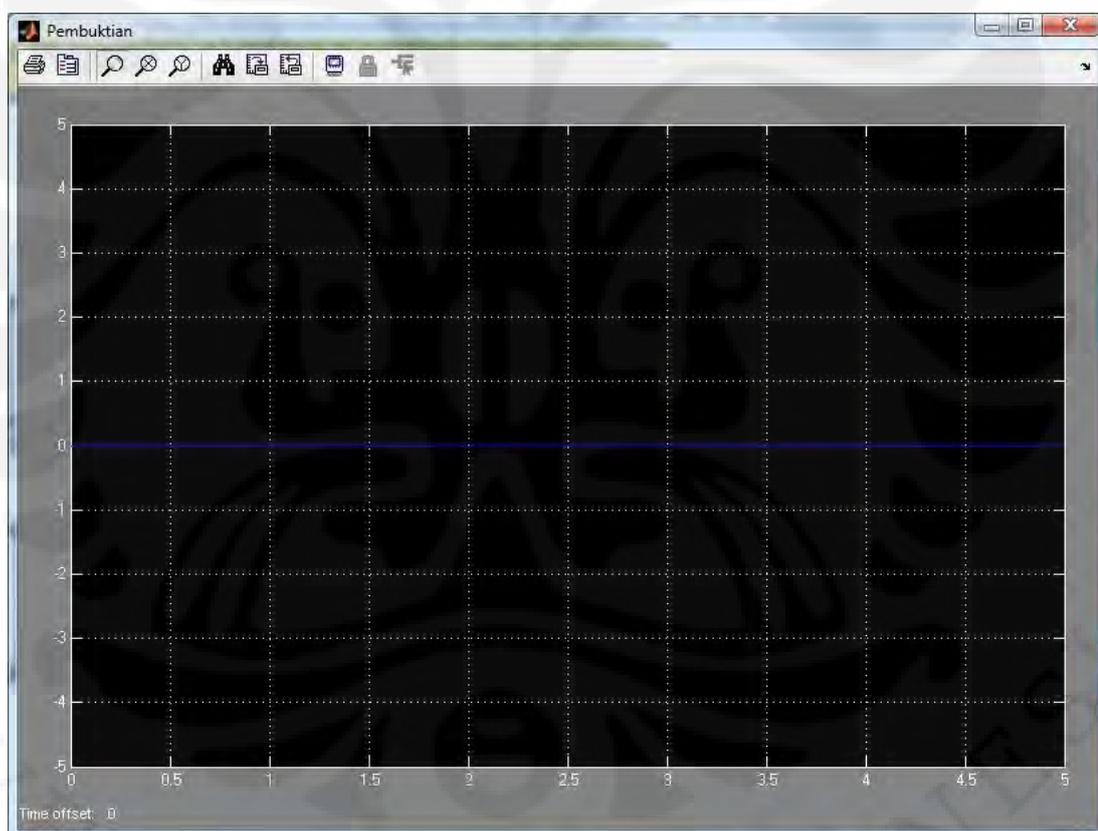
Gambar 4.4 Perbandingan data bit kirim (atas) dan data bit terima (bawah)

Pada Gambar 4.2, Gambar 4.3 dan Gambar 4.4 data baru muncul setelah detik ke 1 karena data yang dihasilkan adalah *frame based*. Data *frame based* akan dikirim apabila jumlah *samples per frame* telah memenuhi konfigurasi yang telah dibuat sehingga untuk mendapatkan jumlah bit sesuai dengan *samples per frame*

yang diinginkan, maka bit-bit data harus disimpan terlebih dahulu sampai jumlah *samples per frame* terpenuhi sehingga terjadi *delay*.

Dari Gambar 4.4 terlihat bahwa data kirim dan data terima memiliki bentuk yang sama. Hal ini menunjukkan bahwa data yang diterima telah sesuai seperti sinyal aslinya, yaitu nilai yang terdapat pada data kirim.

Untuk membuktikan kebenaran nilai data terima terhadap data kirim maka dilakukanlah pengujian data. Pengujian data dilakukan dengan melakukan operasi logical XOR antara data kirim dengan data terima. Dengan membandingkan sinyal yang dikirim oleh *transmitter* dengan sinyal yang diterima di *receiver* yang telah diproses, dapat diketahui bahwa data yang diterima sama dengan data yang dikirim seperti terlihat pada scope pembuktian pada Gambar 4.5.



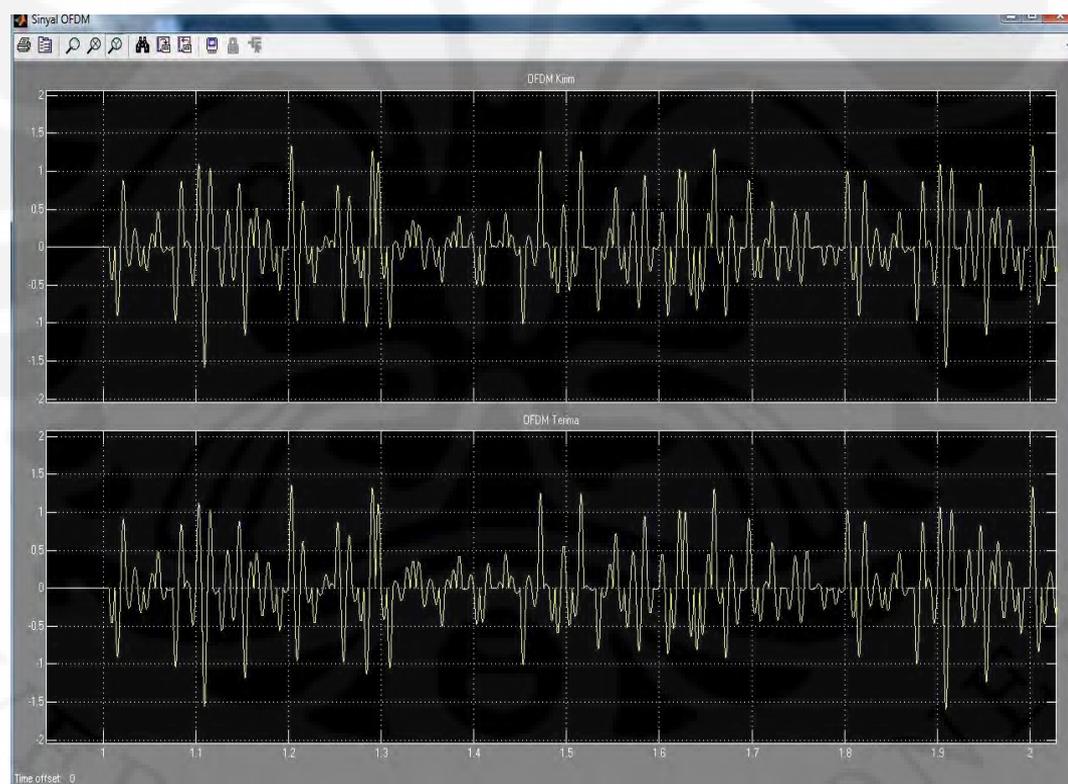
Gambar 4.5 Pembuktian data

Grafik pada *scope* pembuktian selalu bernilai nol karena dilakukan proses XOR terhadap data yang dikirim dengan data yang diterima. Apabila data yang dikirim dengan data yang diterima bernilai sama maka hasil XOR sama dengan nol. Oleh karena data yang dikirim dengan data yang diterima selalu bernilai nol

maka hasil XOR juga akan selalu bernilai nol. Hal ini menunjukkan bahwa rangkaian *baseband* PLC yang telah diuji sudah berfungsi dengan semestinya.

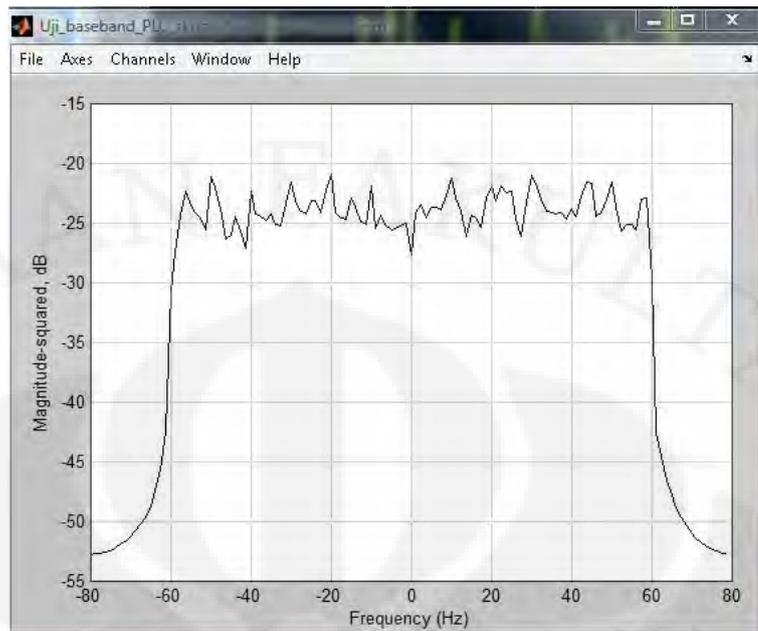
4.1.2. Perbandingan Sinyal dan Spektrum OFDM

Setelah data kirim mengalami proses *encoding* dan modulasi pada sisi *transmitter*. Pada blok OFDM *Modulation* terdapat *sub-blok* IFFT untuk proses pembentukan sinyal OFDM dan transformasi dari domain frekuensi ke domain waktu sehingga bentuk sinyal OFDM setelah melewati IFFT akan terlihat seperti Gambar 4.6 bagian atas. Sinyal keluaran IFFT merupakan sinyal yang dikirim oleh *transmitter* OFDM. Setelah melewati blok AWGN sebagai representasi dari kanal *noise* PLC dengan parameter SNR 30 dB dan *input* daya sinyal 0,01 W, sinyal tersebut kemudian diterima oleh *receiver* OFDM yang selanjutnya akan mengalami proses FFT dan demodulasi. Spektrum sinyal setelah melalui proses FFT dapat dilihat pada Gambar 4.6 bagian bawah.



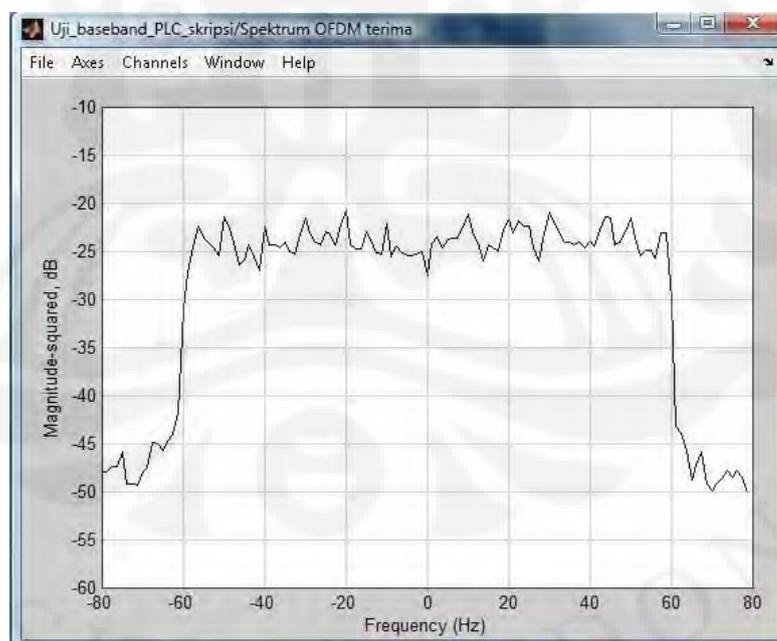
Gambar 4.6 Perbandingan Sinyal OFDM kirim (atas) dan OFDM terima (bawah)

Selain melihat bentuk sinyal OFDM dalam domain waktu, dapat dilihat pula bentuk spektrum frekuensi dari sinyal OFDM. Gambar 4.7 menunjukkan spektrum frekuensi OFDM pada sisi pengirim.



Gambar 4.7 Spektrum OFDM sisi pengirim

Banyaknya data pada satu *frame* sebanyak 84, dengan penambahan pilot dan *zero padding* besar satu *frame* menjadi 128 yang merupakan nilai IFFT. Gambar 4.8 menunjukkan spektrum OFDM pada sisi penerima setelah melewati kanal noise AWGN.



Gambar 4.8 Spektrum OFDM sisi penerima

Spektrum sinyal setelah proses FFT tidak berbeda jauh dengan spektrum sinyal sebelum IFFT. Hal ini berarti sinyal yang ditangkap oleh *receiver* tidak

mengalami perubahan atau kerusakan. Keadaan ini bisa dicapai karena tidak adanya gangguan terhadap sinyal yang dikirim oleh *transmitter*.

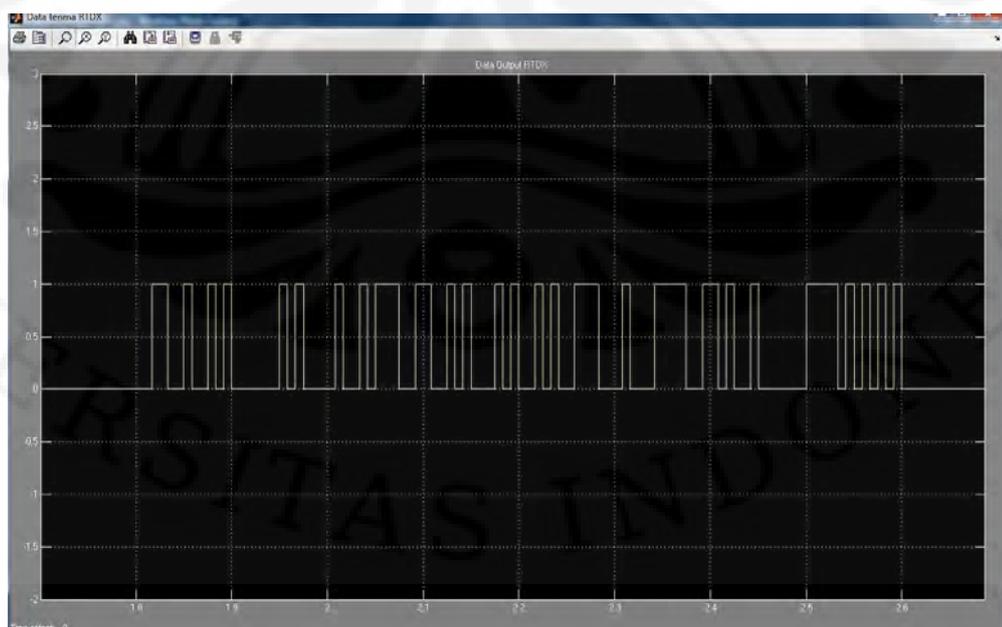
4.2. Hasil Pengujian dengan DSK TMS320C6713

Pengujian dengan menggunakan DSK TMS320C6713 ini bertujuan untuk melakukan simulasi eksternal model rangkaian *baseband* PLC pada sebuah perangkat keras, yaitu papan DSK TMS320C6713. Prosedur simulasi pada papan DSK TMS320C6713 telah dibahas pada bab sebelumnya. Pengujian dilakukan dengan bantuan RTDX untuk mengambil data *output* dari DSK yang akan dibandingkan dengan data hasil pengujian Simulink. Pengujian hanya dilakukan pada data *output* karena RTDX tidak dapat digunakan dalam mengambil data bilangan kompleks seperti data pada OFDM.

4.2.1 Hasil Data Output

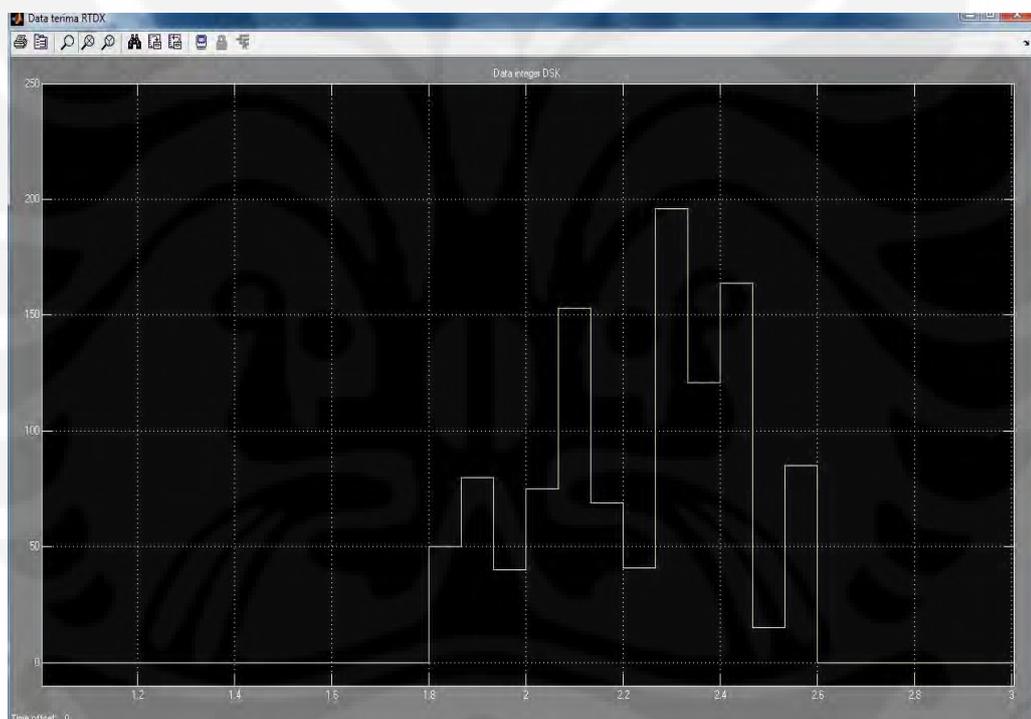
Ujicoba dilakukan dengan menggunakan RTDX karena *output* berupa bit-bit. Dengan menggunakan RTDX maka *output* yang berupa bit-bit akan dikirim ke komputer dan data diolah secara *digital* sehingga tidak mengganggu atau merusak *output*.

Pengujian kebenaran data *output* dari *receiver* dilakukan dengan membandingkan data *output* RTDX dengan data *output* yang didapat dari simulink. *Output* dari rangkaian *baseband* PLC dengan menggunakan RTDX didapatkan data seperti Gambar 4.9.



Gambar 4.9 Data *Output* RTDX dalam bit

Data keluaran dari RTDX diatas masih berupa bilangan biner (*bit stream*) yang terdiri atas “1” dan “0” sama dengan hasil pengujian pada simulink pada Gambar 4.4 bagian bawah. Apabila diperhatikan memang agak sulit untuk membuktikan kebenaran data keluaran RTDX dengan data keluaran dari simulink. Sehingga untuk lebih memperjelas data keluaran RTDX maka dilakukan pengambilan data keluaran RTDX untuk nilai *integer*. Pada data *integer* dapat terlihat ketinggian level data yang berbeda untuk setiap data sampelnya. Level data ini menunjukkan nilai *integer* untuk masing-masing data sampel. Hasil ini sesuai dengan nilai data input pada Tabel 4.1, dimana jumlah sampelnya ada 12 simbol dengan nilai *integer* yang bersesuaian. Nilai data keluaran RTDX untuk bilangan integer dapat dilihat pada Gambar 4.8.



Gambar 4.10 Data Output RTDX dalam integer

Dari kedua gambar tersebut dapat diketahui bahwa data *output* yang didapat dengan menggunakan RTDX pada Gambar 4.8 sama dengan data *output* yang berasal dari simulink pada Gambar 4.3. Data *output* yang berasal dari simulink sama dengan data yang dikirim dari *transmitter*. Oleh karena itu, data *output* yang didapat dari RTDX sama dengan data yang dikirim dari *transmitter* pada Gambar 4.2. Hal ini menunjukkan bahwa simulasi model rangkaian *baseband* PLC yang diujikan sudah berfungsi dengan semestinya.

4.3. Perbandingan Hasil Simulink dan Hasil DSK TMS320C6713

Dari hasil pengujian Simulink dan hasil pengujian DSK TMS320C6713 diperoleh hasil yang sama untuk nilai data *output*. Baik untuk data *output* berupa bit maupun data *output* integer. Perbedaan uji Simulink dengan DSK terletak pada waktu awal munculnya data yang dibangkitkan pada grafik *scope*.

Pada uji Simulink data *output integer* (Gambar 4.3) muncul setelah detik ke 1 sedangkan pada uji DSK data *output integer* (Gambar 4.9) muncul setelah detik ke 1,8. Pada Simulink terjadinya *delay* ini karena data yang diterima melalui blok *unbuffer*, yang akan menyimpan memori terlebih dahulu data yang masuk hingga mencapai 1 *frame* baru kemudian data dikeluarkan. Pada uji DSK *delay* berlangsung lebih lama karena simulasi berlangsung secara eksternal. Selain ada blok *unbuffer* sebelum RTDX pada model Simulink juga terdapat pula blok *unbuffer* pada *scope* untuk menampilkan data *output*.

Hasil data *output* yang sama dari pengujian Simulink dan pengujian pada DSK membuktikan bahwa model simulasi baseband PLC yang dibuat telah berhasil diterapkan pada perangkat DSK TMS320C6713.

BAB V**KESIMPULAN**

Berdasarkan hasil pengujian dan analisis yang dilakukan maka dapat disimpulkan sebagai berikut:

1. Dari hasil pengujian menunjukkan bahwa perancangan model *baseband* PLC dapat diterapkan dengan DSK TMS320C6713 dimana pembuatan model berbasiskan simulink.
2. Hasil perbandingan antara data kirim dan data terima pada uji Simulink dan DSK telah menghasilkan hasil yang sama sesuai dengan tujuan dari skripsi ini.
3. Pada proses pengambilan data Simulink terdapat penundaan selama 1 detik. Hal ini karena memori harus menyimpan sementara semua datanya pada 1 *frame* yang berisi 12 sampel data selama 1 detik baru kemudian ditampilkan.

DAFTAR REFERENSI

- [1] “___”, “*TES & TEKNOLOGI Powerline Communication (PLC) - Jaringan via Listrik - CHIP Online Indonesia*”, 2007. Diakses dari: <http://www.chip.co.id/tips-and-technologies/tes-teknologi-powerline-communication-plc-jaringan-via-listrik.html> , 14 September 2009
- [2] Zyren, Jim, “*HomePlug 1.0 PHY for Smart Grid and Electric Vehicle Applications*”, Atheros Communications, 2010. Diakses dari: http://www.atheros.com/pt/plc/downloads/whitepaper_HomePlug.PHY.for.Smart.Grid.pdf, Mei 2010
- [3] “___”, PLC G3 Physical Layer Specification, ERDF. <http://www.maxim-ic.com/products/powerline/pdfs/G3-PLC-Physical-Layer-Specification.pdf> , 3 Mei 2010
- [4] Lee, M. K, *HomePlug 1.0 powerline communication LANs-protocol description and performance results*, International Journal of Communication System, 2003 <http://www.list.ufl.edu/publications/IJCS%20HP%201%5B1%5D.0%20LAN%20Protocol.pdf>
- [5] M.Deinzer and M.Stoger, “*Integrated PLC-modem based on OFDM*,”in Proc.3rdInt.Symp.Power-LineCommunications, Lancaster,Mar.30–Apr. 1999. Diakses dari: <http://www.iad-de.com/plcsym/plcpaper99.pdf> ,
- [6] Langton, Charan, “*Orthogonal Frequency Division Multiplexing*”, OFDM tutorial, 2004. Diakses dari: <http://www.complextoreal.com/chapters/ofdm2.pdf>, Mei 2010
- [7] Sigit Puspito W.J., “*Mengenal Teknologi Orthogonal Frequency Division Multiplexing (OFDM) pada Komunikasi Wireless*”, Elektro Indonesia, Nomor 24, Tahun V, Januari 1999. Diakses dari: <http://www.elektroindonesia.com/elektro/tel24.html> , 20 Mei 2010
- [8] Poncius S, Pontas, *Rancang Bangun Rangkaian Penerima OFDM Dengan Menggunakan DSK TMS320C6713 Berbasis Simulink*. Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2008
- [9] “___”, *Introduction to Simulink, Link for CCS & Real-Time Workshop*,2006. Diakses dari: <http://www.emba.uvm.edu/~mirchand/classes/EE275/2007/Real-Time/Lab6.pdf> , 8 Februari 2010
- [10] “___”, Phase Shift Keying, diakses dari: http://en.wikipedia.org/wiki/Phase-shift_keying , 17 Mei 2010.

- [11] “_____”, Power Line Communication, diakses dari:
http://en.wikipedia.org/wiki/Power_line_communication, 17 Mei 2010
- [12] Moreira, J.C. dan Farrell, P.G., *Essentials of Error-Control Coding*, John Wiley & Sons Ltd, Chichester, 2006.
- [13] Sweeney, Peter, *Error Control Coding From Theory to Practice*, Jo John Wiley & Sons Ltd, Chichester, 2002.

DAFTAR PUSTAKA

Matlab help, *signal processing toolbox*, mathworks inc

Matlab help, *communication blockset*, mathworks inc

Ishak, Maulana. *Rancang Bangun Baseband WIMAX dengan Menggunakan DSK TMS320C6713 Berbasis Matlab Simulink*. Skripsi , Program Sarjana Fakultas Teknik UI, Depok, 2009.

Schulze, Henrik, Christian Luders, *Theory and Application of OFDM and CDMA* (Chichester: John Wiley & Sons, 2005)

Chassaing, Rulph, "Digital Signal Processing and Applications with the C6713 and C6416 DSK," (New Jersey: John Wiley & Sons, Inc, 2005)

LAMPIRAN

1. Program *Driver* RTDX

```

function RTDXdriver(modelname)
% RTDXDRIVER Reads and plots data through an RTDX channel.

[modelpath,modelname,modelext] = fileparts(modelname);

cc = ccstdsp;
set(cc,'timeout',50);
if ~isrtdxcapable(cc)
    error('Processor does not RTDX support');
end

cc.reset; pause(1);
cc.cd(modelpath);
cc.visible(1);

open(cc,sprintf('%s.pjt',modelname));
load(cc,sprintf('%s.out',modelname));

rx = cc.rtdx;
rx.set('timeout', 100); % Reset timeout = 10 seconds
rx.configure(64000,2);
rx.open('pesan_akhir','r');
rx.enable; % enable RTDX
cc.run; % cc.enable can be placed here
pause(1); % cc.enable cannot be placed here; too much time had
passed
    % RTDX processing will be 'stalled'
% source array preparing
ukuran=25;
i=1;
enable(rx,'pesan_akhir');
while (i<ukuran)
    if isenabled(rx,'pesan_akhir')

pesanakhir(i,1)=readmsg(cc.rtdx,'pesan_akhir','double');
        end
        i=i+1;
end

RTDXcleanup(cc,rx);
    matfile=strcat('pesanujung','.mat');
    save(matfile,'pesanakhir');

%=====
% Put RTDX back to good state
%=====

function RTDXcleanup(cc,rx)
if isrunning(cc), % if the target DSP is running
    halt(cc); % halt the processor
end
    cc.reset;
disable(rx,'pesan_akhir');
disable(rx); % disable RTDX
close(cc.rtdx,'pesan_akhir');

```