

# Implementasi *Hybrid DS/FH Spread Spectrum* menggunakan DSK TMS302C6416T

Aji Wisnu Laksita, Suwadi, dan Titiek Suryani

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: aji\_laksita@rocketmail.com, suwadi110@gmail.com, titiks@ee.its.ac.id

**Abstrak**—*Spread spectrum* adalah sebuah metode komunikasi dimana semua sinyal komunikasi disebar di seluruh spektrum frekuensi yang tersedia. Sinyal informasi akan ditebarkan pada seluruh pita frekuensi yang disediakan dan dilakukan secara acak. Metode *spread spectrum* merupakan metode komunikasi yang memiliki ketahanan terhadap *jamming*.

*Hybrid DS/FH spread spectrum* adalah sistem komunikasi *spread spectrum* yang menggabungkan antara sistem komunikasi *direct sequence spread spectrum* (DSSS) dan *frequency hopping spread spectrum* (FHSS). *Hybrid DS/FH spread spectrum* dapat diprogram dengan menggunakan DSP processor. DSK TMS320C6416T adalah suatu hardware untuk memproses sinyal yang termasuk dalam prosesor TMS320 produksi *Texas Instrument*. Sistem komunikasi *hybrid DS/FH spread spectrum* diharapkan memiliki ketahanan yang lebih terhadap gangguan seperti kondisi kanal AWGN dan *jamming* yang berupa *single tone jamming* dan *multitone jamming*.

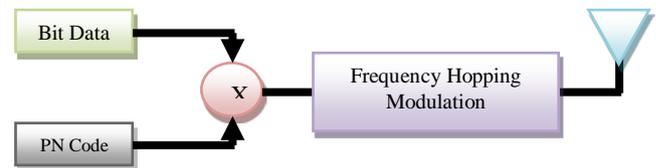
Pada kondisi kanal AWGN yang buruk dengan tingkat *signal to noise ratio* (S/N) rendah dengan  $E_b/N_0$  0 dB, sistem komunikasi *hybrid DS/FH spread spectrum* memiliki nilai *bit error rate* (BER) 0.04. *Hybrid DS/FH spread spectrum* memiliki kehandalan yang sangat baik terhadap *jamming* yang berupa *single tone jamming* dan *multitone jamming* terbukti pada pengujian kehandalan terhadap pengaruh *single tone jamming* dengan daya *tone* dua kali daya sinyal *carrier* sistem dan melewati kanal AWGN dengan  $E_b/N_0$  10 dB, sistem komunikasi *hybrid DS/FH spread spectrum* memiliki BER 0.00001 dan 0.03 untuk *multitone jamming* dengan parameter yang sama. Berdasarkan uraian tersebut dapat ditarik kesimpulan bahwa sistem komunikasi *hybrid DS/FH spread spectrum* tahan terhadap gangguan seperti kondisi kanal AWGN dan *jamming* yang berupa *single tone jamming* dan *multitone jamming*.

Kata kunci—*spread spectrum*, *direct sequence*, *frequency hopping*, *hybrid DS/FH spread spectrum*, *bit error rate*, *single tone jamming*, *multitone jamming*, DSK TMS320C6416T

## I. PENDAHULUAN

Dalam bidang tertentu khususnya di bidang militer, kerahasiaan sebuah informasi merupakan hal yang wajib terpenuhi. Oleh karena itu, dibutuhkan teknik telekomunikasi yang tahan terhadap bermacam-macam gangguan. Gangguan tersebut dapat berupa interferensi, *jamming* bahkan penyadapan. Teknik *spread spectrum* merupakan teknik yang dapat memberikan jaminan aman terhadap informasi yang dikirimkan. Pada teknik *spread spectrum* khususnya *hybrid DS/FH spread spectrum*, informasi akan dikirimkan secara acak dan tersebar pada *band-band* frekuensi tertentu. Hal itu mengakibatkan informasi yang dikirimkan akan sangat sulit untuk dilacak/disadap serta tahan terhadap interferensi dan *jamming*.

*Hybrid DS/FH spread spectrum* adalah sebuah teknik *spread spectrum* yang menggabungkan antara teknik *direct sequence spread spectrum* dengan *frequency hopping*



Gambar 1. Transmitter *hybrid DS/FH spread spectrum*

*spread spectrum* dengan tujuan meningkatkan kehandalan sistem. Teknik *hybrid DS/FH spread spectrum* ini akan diimplementasikan pada sebuah DSP starter kit (DSK) TMS320C6416T dengan menggunakan integrasi perangkat lunak *simulink* MATLAB. Dengan diimplementasikannya teknik ini diharapkan akan tercipta sistem yang handal khususnya tahan terhadap *jamming*.

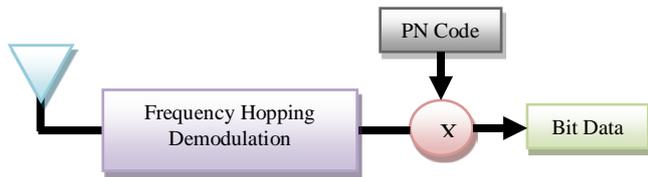
## II. TEORI PENUNJANG

### A. *Frequency-Hopping Spread Spectrum* (FHSS)

*Hybrid DS/FH spread spectrum* merupakan teknologi penggabungan antara *direct sequence* dengan *frequency hopping*. Penggabungan ini bertujuan untuk meningkatkan keamanan dari data informasi yang dikirimkan. Teknologi *hybrid* ini tidak hanya dapat mengacak dan menyebarkan data informasi yang akan dikirimkan akan tetapi juga melompat-lompatkan informasi tersebut pada frekuensi yang telah ditentukan. [2]

Seperti terlihat pada Gambar 1, data yang akan dikirimkan terlebih dahulu akan dikalikan dengan kode PN sehingga menghasilkan data acak yang tersebar untuk setiap data informasi. Pada modulasi *frequency hopping*, data acak yang diterima kemudian akan di lompat-lompatkan pada frekuensi pembawa tertentu sesuai dengan pola lompatan dan kemudian dikirimkan. Pada dasarnya perbedaan sistem *hybrid DS/FH spread spectrum* dengan *frequency hopping spread spectrum* terletak pada bit data yang dilompatkan. Pada *frequency hopping spread spectrum* yang dilompatkan adalah bit data informasi sedangkan pada *hybrid DS/FH spread spectrum* yang dilompatkan adalah *chips* atau data acak yang memiliki laju data lebih tinggi daripada data informasi.

Pada bagian penerima seperti terlihat pada Gambar 2, sinyal informasi yang di terima akan didemodulasi menggunakan *frequency hopping modulator* untuk memisahkan data dengan frekuensi pembawanya. Hasil dari demodulasi menghasilkan data acak (*chips*). Untuk mengembalikan data acak menjadi data informasi yang sebenarnya dilakukan dengan cara mengalikan data acak tersebut dengan kode PN yang sama digunakan pada pengirim.



Gambar 2. Receiver hybrid DS/FH spread spectrum

**B. Kode PN**

Kode PN atau *pseudorandom noise code* adalah kode yang berisi data berupa bilangan *biner sequence*. *Pseudorandom noise* memiliki satuan *chips* yang dibangkitkan secara acak dan terus menerus namun memiliki pola tertentu. Karena kode ini memiliki pola tertentu maka disebut *pseudorandom* atau acak semu sehingga dapat diketahui secara pasti oleh pengirim dan penerima.

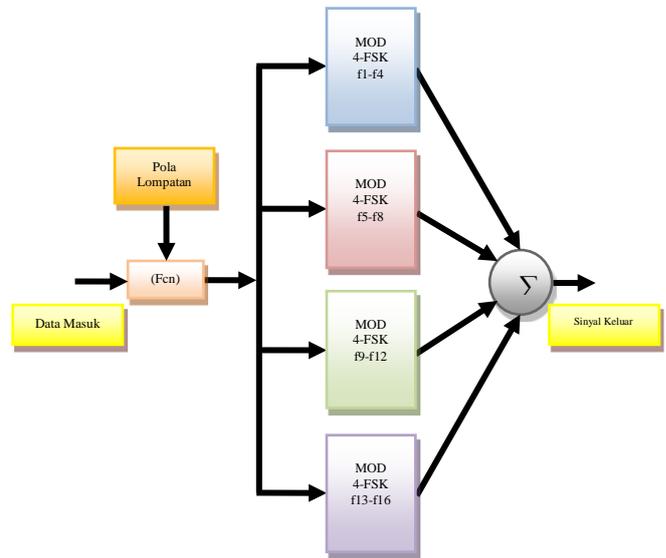
Seperti halnya pada sistem komunikasi *direct sequence spread spectrum*, kode PN yang digunakan pada sistem komunikasi *hybrid DS/FH spread spectrum* juga memiliki *bit rate (chips rate)* yang lebih tinggi daripada *bit rate* data informasi. Oleh karena itu kode PN tidak hanya berfungsi untuk mengacak data informasi tetapi juga menyebarkannya.

**C. Modulasi FHMFSK**

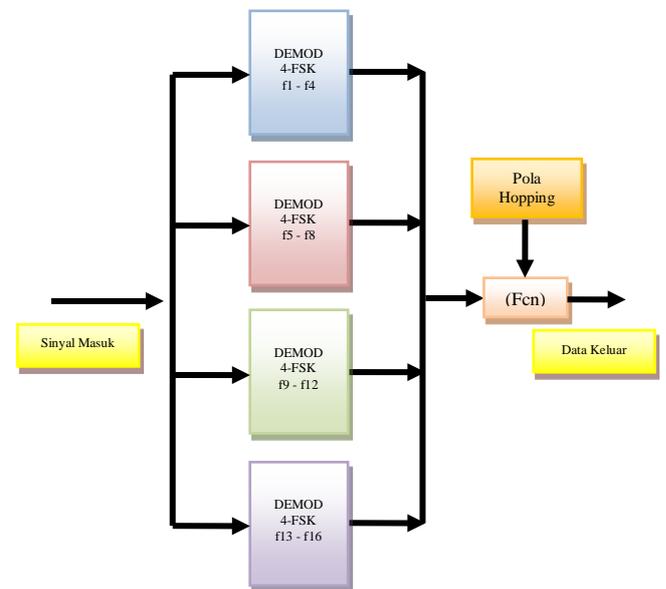
Modulasi FHMFSK atau *frequency hopping m-ary frequency shift keying* merupakan sebuah modulasi FSK yang telah dimodifikasi secara khusus. [2] Pada modulasi FSK data informasi akan dikalikan dengan beberapa sinyal carrier dengan frekuensi tertentu tergantung dari *mapping* simbol data informasi. Jadi untuk membedakan kode/symbol data informasi yang satu dengan yang lainnya yaitu dengan membedakan frekuensi pada sinyal *carrier*.

Modifikasi yang dilakukan pada modulasi 4-FSK adalah dengan menambahkan pola *hopping* dan menggandakan modulasi 4-FSK dalam satu model blok modulasi. Pola *hopping* berfungsi untuk menentukan pola lompatan pada *range* frekuensi (submodulasi 4-FSK) tertentu. Dalam modulasi FH 4-FSK seperti terlihat pada Gambar 3, terdapat 4 submodulasi 4-FSK dengan frekuensi yang berbeda-beda yang digunakan sebagai level lompatan. Setiap level lompatan terdapat 4 frekuensi pembawa (*carrier*) yang berbeda-beda. Jadi secara keseluruhan terdapat 16 frekuensi pembawa berbeda yang digunakan. Pemilihan frekuensi pembawa yang digunakan tergantung dari laju data (*data rate / symbol rate*) informasi yang akan dimodulasi. Sampel frekuensi yang digunakan minimal 2 kali frekuensi pembawa tertinggi yang digunakan untuk memenuhi teorema *nyquist* agar tidak terjadi *aliasing*.

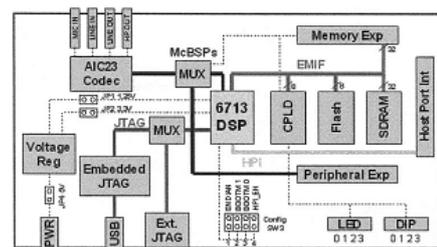
Pada bagian demodulator seperti terlihat pada Gambar 4, data akan di pisahkan dari frekuensi pembawa dengan mengkalikan sinyal informasi dengan frekuensi pembawa. Data hasil perkalian tersebut masih berupa data acak yang bercampur dengan data pola lompatan, oleh karena itu data tersebut perlu dipisahkan untuk menghasilkan data acak yang sebenarnya dengan menggunakan pola hopping yang digunakan pada modulasi pemancar.



Gambar 3. Modulasi FH 4-FSK



Gambar 4. Demodulasi FH 4-FSK



Gambar 5. Diagram blok DSK TMS320C6416T [5]

**D. DSP Starter Kit TMS320C6416T**

*Texas Digital Signal Processor (DSP)* seri TMS320C6x adalah mikroprosesor berkecepatan tinggi dengan tipe arsitektur yang cocok digunakan untuk mengolah sinyal. Notasi C6x merupakan kode dari produk DSP keluaran Texas Instruments TMS320C6000. Dengan menggunakan arsitektur *very long instruction word (VLIW)*, DSP C6x menjadi prosesor tercepat keluaran Texas Instruments. Arsitektur VLIW pada DSP C6x sangat cocok untuk proses perhitungan yang intensif. [4] Diagram blok DSK TMS320C6416T dapat dilihat pada Gambar 5.

DSP dikemas dalam sebuah DSK (*Digital Signal Processor Starter Kit*) dengan ukuran sekitar 5 x 8 inch

dimana terdapat C6416 *floating-point* dan sebuah 32 bit *stereo codec* TLV320AIC23 (AIC23) sebagai masukan dan keluaran. *Codec* AIC23 menyediakan ADC dan DAC dengan clock 12 MHz dan sampling rate 8 – 96 kHz. DSK memiliki memori tipe *synchronous dynamic random access memory* (SDRAM) sebesar 16 MB dan flash memori sebesar 256 kB. Ada 4 konektor pada DSK yang berfungsi sebagai masukan dan keluaran yaitu:

- MIC IN untuk masukan *microphone*.
- LINE IN untuk masukan *line*.
- LINE OUT untuk keluaran *line*.
- HEADPHONE untuk keluaran *headphone* yang tergabung dalam LINE OUT.

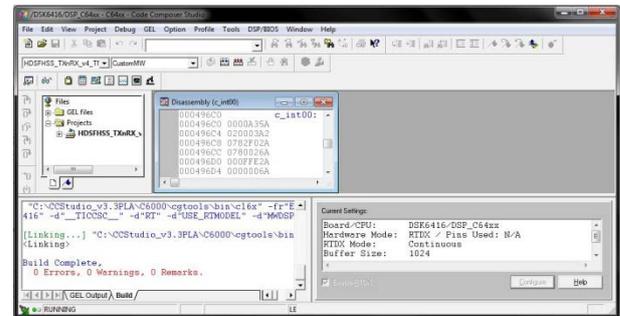
*Dip switches* dalam DSK dapat difungsikan sesuai program dan menyediakan fungsi kontrol. Selain itu terdapat pengatur tegangan yang menyediakan 1,26 volt untuk prosesor C6713 dan 3,3 volt untuk sistem memori dan kelengkapan lain. Fitur-fitur lain yang tersedia dalam DSK TMS320C6713 terdapat dalam lampiran.

#### E. Code Composer Studio v3.3

*Code composer studio* (CCS) merupakan perangkat lunak yang digunakan untuk menghasilkan kode seperti *C compiler*, *assembler* dan *linker* untuk DSK keluaran Texas Instruments. CCS memiliki kemampuan *real-time debugging*. *C compiler* mengkompilasi sebuah program dalam bahasa C dengan ekstensi \*.c, sedangkan untuk menghasilkan file *assembly* menggunakan ekstensi \*.asm. *Assembler* memproses file \*.asm untuk dihasilkan file bahasa mesin dengan ekstensi \*.obj. Kemudian *linker* menggabungkan file – file tersebut menjadi *executable file* dengan ekstensi \*.out. File ini dapat dimasukkan ke dalam prosesor C6416. DSK harus dihubungkan ke PC melalui port USB agar bisa diprogram dengan bantuan CCS. [7]

CCS juga mendukung kegiatan *debugging* antara lain, setting *breakpoint*, secara otomatis meng-*update* jendela saat *breakpoint*, mengamati nilai variabel, melihat dan mengamati memori dan register, menggunakan *probe point* untuk mengalirkan data dari dan ke target untuk mengumpulkan *snapshot memory*, menggambarkan sinyal yang ada pada target, melakukan *profiling* terhadap statistik eksekusi, memeriksa instruksi C dan instruksi yang di-*disassembly* pada target. [7] Gambar 6 menunjukkan tampilan CCS v3.3.

MATLAB telah menyediakan sebuah fungsi untuk berkomunikasi dengan DSK TMS320C6x dengan bantuan CCS. Semua perangkat lunak ini membangun dan memvalidasi hasil perancangan pengolahan sinyal digital mulai dari konsep sampai dengan menghasilkan kode yang akan dijalankan oleh DSK TMS320C6x. CCS memiliki fitur untuk *editing*, *building*, *debugging*, *code profiling* dan *project management*. CCS mengintegrasikan simulasi yang sudah dibentuk dari Simulink Matlab dan mengkonversikannya ke dalam bahasa C dan *assembly*. File keluaran dari proses ini di-*download* sebagai keluaran pada DSK TMS320C6713. Proses *running* dapat diakses dari *CCS debugging tools* atau melihat langsung hasil simulasi Simulink Matlab.



Gambar 6. Tampilan CCS v3.3

### III. PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada model sistem komunikasi *Hybrid DS/FH spread spectrum* terdiri dari 2 blok yaitu blok *transmitter* dan *receiver*. Masing-masing blok tersebut dibagi menjadi beberapa sub-blok yang terdiri dari blok *direct sequence*, *frequency hopping* dan modulator/demodulator.

#### A. Transmitter

*Direct sequence* pada *transmitter* berfungsi untuk mengacak data informasi masukan. Data masukan disini adalah berupa biner 0 dan 1 keluaran dari blok "*Random Integer Generator*". Untuk mengacak data tersebut dapat dilakukan dengan cara mengkalikan (*cross product*) dengan kode PN yang di keluarkan oleh blok "*PN Sequence Generator*". Ada beberapa parameter yang harus di perhatikan dalam pengacakan data. Yang pertama adalah bahwa *bit rate* PN code (*chip rate*) harus lebih tinggi daripada *bit rate* data. Dalam hal ini, *chip rate* kode PN adalah 8 *bit/s* dan untuk *data rate* 1 *bit/s*. Yang kedua adalah sebelum dikalikan, data dan kode PN harus dirubah terlebih dahulu menjadi bilangan *bipolar* (1 dan -1). Setelah proses perkalian (*cross product*) selanjutnya dapat dirubah kembali menjadi bilangan *unipolar* (0 dan 1). Untuk menyamakan *clock rate* antara data dan kode PN diperlukan blok "*Rate Transition*" pada keluaran blok "*Random Integer Generator*".

Bagian *frequency hopping* pada sistem berfungsi untuk melompatkan *chips* atau *bit* data yang telah diacak pada frekuensi tertentu. Lompatan frekuensi tersebut tidaklah acak tetapi memiliki pola tertentu. Pola lompatatan tersebut dibagi menjadi 4 level dan setiap level memiliki 4 frekuensi yang berbeda sehingga secara keseluruhan ada 16 frekuensi yang berbeda.

Pola tersebut adalah keluaran dari blok "*Repeating Sequence Stair*". Pola tersebut dapat dirubah-rubah sesuai keinginan. Dalam hal ini pola lompatan yang di pakai adalah 3-1-4-2 dan akan diulang secara terus-menerus. Dari pola lompatan tersebut dapat diketahui lompatan pertama menduduki level 3, selanjutnya akan melompat ke level 1 dan seterusnya. Untuk menentukan pada frekuensi berapa yang diduduki tergantung dari data masukan atau dalam hal ini *bit* keluaran *direct sequence*. Karena ada 4 frekuensi yang berbeda pada setiap levelnya maka *bit* keluaran *direct sequence* perlu dirubah menjadi bilangan *integer* menggunakan blok "*Bit to Integer Converter*" dengan parameter 2 *bit/symbol* sehingga akan menghasilkan bilangan dengan *range* dari 0 sampai 3. Untuk memudahkan dalam memproses bilangan tersebut maka *range*-nya harus dinaikkan menjadi 1 sampai 4 dengan cara menambahkannya dengan angka 1.

Dari penjelasan diatas maka didapatkan level lompatan dengan *range* dari 1 sampai 4 dan data masukan (*integer*) memiliki *range* 1 sampai 4. Selanjutnya, untuk menentukan frekuensi ke berapa yang diduduki dapat menggunakan persamaan (1) dibawah ini.

$$\text{Kedudukan Frekuensi} = ((\text{Level Lompatan} - 1) * 4) + \text{Data Masukan} \quad (1)$$

Berdasarkan persamaan (1) di atas maka akan didapat bilangan dengan *range* dari 1 sampai 16 dan bilangan tersebutlah yang menunjukkan frekuensi ke berapa yang diduduki. Rumus di atas akan dimasukkan kedalam blok “*Embedded MATLAB Function*”.

*Input* modulasi adalah bilangan integer dengan *range* dari 1 sampai 16. Ada 16 frekuensi berbeda yang digunakan dari  $f_1$  sampai  $f_{16}$ . Untuk menentukan frekuensi minimal yang digunakan adalah dengan melihat data rate masukan modulasi. *Bit rate* keluaran dari blok *direct sequence* adalah 8 bit/s, kemudian dirubah menjadi bilangan integer dengan parameter 2 bit/symbol maka terjadi penurunan data rate menjadi 4 bit/s dan bertahan sampai masuk modulasi. karena data rate masukan 4 bit/s maka untuk menghindari *aliasing*, frekuensi minimal ( $f_1$ ) yang di gunakan adalah 4 Hz dan untuk fekuensi selanjutnya adalah kelipatan dari 4 yaitu 8 Hz ( $f_2$ ), 16 hz ( $f_3$ ) dan seterusnya sampai 64 Hz ( $f_{16}$ ). Untuk menentukan sample time yang dibutuhkan dapat melihat dari frekuensi maksimal yang digunakan, dalam hal ini frekuensi maksimal yang digunakan adalah 64 Hz. Untuk memenuhi aturan *nyquist* maka untuk satu periode sinyal setidaknya perlu 2 sampel agar tidak terjadi *aliasing*. Untuk membentuk satu periode sinyal yang sempurna dalam hal ini menggunakan 16 *sample/period*. Karena frekuensi maksimal yang digunakan adalah 64 Hz (64 *period/s*) maka perlu 1024 *sample/s*.

Blok “*Rate Transition*” berfungsi untuk menyamakan *clock* atau *data rate* data masukan dengan blok “*Hopping Mod*”. Blok “*Hopping Mod*” tersebut merupakan sebuah blok *subsystem* dari modulasi. Di dalam blok *subsystem* tersebut terdapat beberapa blok, sebagai pertama adalah blok “*Lookup Table*” yang berfungsi sebagai *mapping*. Blok yang kedua adalah “*Cosine Wave*” berfungsi untuk membangkitkan sinyal pembawa (*carrier*). Selanjutnya adalah blok “*Product*” dan “*Add*” yang berfungsi untuk perkalian dan penjumlahan. Sinyal pembawa diurutkan dari paling atas berada pada frekuensi 4 Hz ( $f_1$ ), 8 Hz ( $f_2$ ), 16 Hz ( $f_3$ ) dan seterusnya sampai paling bawah yaitu 64 Hz ( $f_{16}$ ).

## B. Receiver

*Demodulator* untuk memisahkan atau menterjemahkan data yang dibawa oleh sinyal pembawa (*carrier*). Untuk melakukannya dapat dilakukan dengan cara mengkalikan sinyal yang diterima dengan masing-masing sinyal pembawa (*carrier*) yang digunakan pada *modulator*. Karena sinyal yang diterima adalah bilangan *complex* maka sebelum dikalikan dengan sinyal pembawa sinyal tersebut harus dirubah terlebih dahulu menjadi bilangan real dengan menggunakan blok “*Complex to Real-Imag*”. Bilangan yang dikalikan tersebut merupakan *sample* dari sinyal yang diterima, dalam hal ini adalah *sample* sinyal pembawa yang di bangkitkan oleh *modulator* yang banyaknya 1024 *sample/s*. Karena data memiliki *symbol rate* 4 *symbol/s* maka hanya di ambil sampel sebanyak 256 *sample* yang akan dijumlahkan menggunakan blok “*Integrate and Dump*”. Hasil dari penjumlahan akan dikalikan dengan *jamming correction*. *Jamming correction* berfungsi sebagai

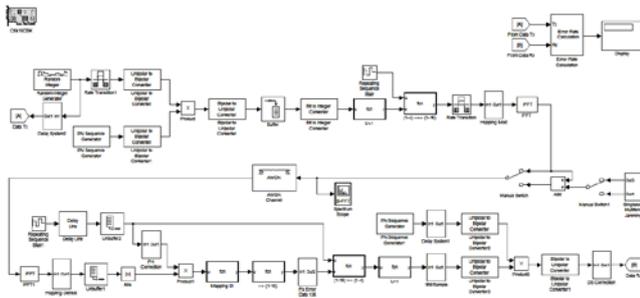
perbaikan akibat adanya *jamming*. Prinsip kerja *jamming correction* tersebut adalah dengan memanfaatkan pola lompatan yang digunakan. Cara kerjanya adalah dengan memilih 4 frekuensi dimana level lompatan berada (1 level lompatan ada 4 frekuensi), dan selain frekuensi tersebut akan diberi nilai 0. Selanjutnya dari semua nilai tersebut akan dibandingkan untuk mencari nilai paling tinggi. Nilai tertinggi akan bernilai 1 sedangkan yang lainnya akan bernilai 0. Jadi pada tahap ini akan ada 1 nilai yang bernilai 1 dan 15 nilai yang bernilai 0 (total ada 16). Nilai-nilai tersebut akan menjadi masukan *korelator*. Keluaran dari *korelator* adalah bilangan *integer* dengan *range* dari 1 sampai 16.

Bagian *frequency hopping* pada *receiver* memiliki fungsi yang berkebalikan dengan yang ada di *transmitter*. Pada bagian *transmitter* berfungsi untuk membangun pola lompatan sedangkan pada *receiver* berfungsinya untuk mengembalikan data menjadi data biner 0 dan 1. Keluaran dari *demodulator* merupakan bilangan *integer* dengan *range* dari 1 sampai 16 yang merupakan kedudukan frekuensi. Untuk mengembalikan datanya maka diperlukan pola lompatan yang digunakan pada *transmitter*. Pola lompatan merupakan bilangan *integer* dengan *range* dari 1 sampai 4 yang dikeluarkan oleh blok “*Repeating Sequence Stair*”. Selanjutnya kedua data tersebut akan diproses menggunakan persamaan (2) dibawah ini.

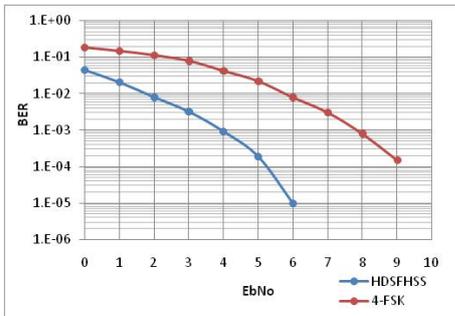
$$\text{Data Keluaran} = \text{Kedudukan Frekuensi} - ((\text{Level Lompatan} - 1) * 4) \quad (2)$$

Persamaan (2) di atas adalah kebalikan dari persamaan (1) yang digunakan pada *transmitter*. Persamaan (2) tersebut akan dimasukkan kedalam blok “*Embedded MATLAB Function ((1-16)>>(1-4))*”. Data keluaran dari proses diatas masih berbentuk bilangan *integer* dengan *range* dari 1 sampai 4 maka perlu di ubah menjadi data biner 0 dan 1. Untuk merubahnya perlu menggunakan blok “*Integer to Bit Converter*”, akan tetapi keluaran dari *converter* tersebut berupa *matrix* yang berisi 2 data biner dan memiliki data rate 4 data (*matrix*) per detik padahal seharusnya memiliki *data rate* 8 bit/s. Oleh karena itu *matrix* tersebut perlu diurai dengan cara membuat *subsystem converter*. Keluaran dari *subsystem* tersebut sudah berbentuk deretan data biner dengan *data rate* 8 bit/s.

Seperti halnya *frequency hopping* pada *receiver*, *direct sequence* pada *receiver* memiliki fungsi yang berkebalikan dengan *direct sequence* pada *transmitter*. Untuk mengembalikan data yang telah di acak menjadi data informasi maka data acak keluaran dari *frequency hopping* harus dikalikan dengan kode PN. Sebelum di kalikan maka data yang berupa biner *unipolar* 0 dan 1 menjadi *bipolar* 1 dan -1 dengan menggunakan blok “*Unipolar to Bipolar Converter*”. Data hasil perkalian yang berupa bilangan biner *bipolar* 1 dan -1 selanjutnya dirubah kembali menjadi bilangan biner *unipolar* 0 dan 1. Bilangan biner tersebut memiliki bit rate 8 bit/s sedangkan data informasi memiliki data rate 1 bit/s, sehingga harus ditambahkan sebuah *converter*. Jika menggunakan hanya memakai blok “*Rate Transition*” maka simbol data yang diambil adalah bit pertama pada deretan chips (8 bit/s). masalah timbul ketika terjadi kesalahan pada bit pertama. Oleh karena itu untuk mengurangi kesalahan maka dibutuhkan blok koreksi yang dalam hal ini adalah sebuah *subsystem* “*DS Correction*”. “*DS Corrector*” tidak hanya sebagai *converter* tetapi juga dapat melakukan koreksi jika terjadi error pada chips yang membangun sebuah simbol atau *bit* data. Blok *subsystem* ini



Gambar 7. Hybrid DS/FH spread spectrum



Gambar 8. BER hybrid DS/FH spread spectrum dan 4-FSK pada kanal AWGN

membandingkan jumlah bit 0 dan 1 dalam deretan 8 bit (chips). Simbol atau bit data yang dipilih adalah nilai chips (0 atau 1) yang paling sering muncul pada deretan chips tersebut (8 bit/s) jadi bukan bit atau chips yang pertama muncul.

Gambar 7 merupakan model sistem komunikasi hybrid DS/FH spread spectrum secara keseluruhan. Sinyal yang dikirim pada transmitter akan dilewatkan pada kanal AWGN dan penambahan jamming yang berupa single tone jamming dan multitone jamming kemudian akan diterima receiver.

Blok sistem yang telah dibuat pada Simulink MATLAB kemudian diimplementasikan menggunakan DSK TMS320C6416T dengan bantuan software CCS v3.3.

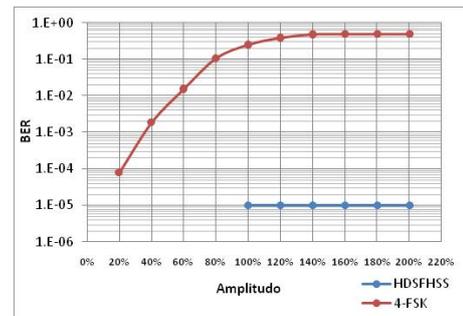
#### IV. PENGUJIAN SISTEM

Pengujian blok sistem komunikasi hybrid DS/FH spread spectrum pada simulink MATLAB dilakukan dengan mengamati nilai bit error rate (BER) pada sistem. Pengujian dilakukan dengan mengubah parameter pada kanal AWGN dan penambahan jamming berupa single tone jamming dan multitone jamming dan dibandingkan dengan sistem komunikasi dengan modulasi 4-FSK dengan parameter kanal dan jamming yang sama. Diambil sebagai pembanding berupa sistem komunikasi dengan modulasi 4-FSK karena memiliki sistem modulasi yang hampir sama dengan sistem komunikasi hybrid DS/FH spread spectrum.

##### A. Pengaruh Kanal AWGN

Pengujian yang dilakukan terhadap sistem komunikasi hybrid DS/FH spread spectrum dengan merubah parameter S/N (signal to noise ratio) pada kanal AWGN dengan EbNo dari 0 dB sampai 10 dB. Sebagai pembanding, juga dilakukan pengujian terhadap sistem komunikasi dengan modulasi 4-FSK dengan parameter yang sama.

Berdasarkan grafik pada Gambar 8 dapat dilihat perbedaan kehandalan atau kinerja antara sistem hybrid DS/FH spread spectrum dengan sistem komunikasi dengan modulasi 4-FSK pada kanal AWGN. Pada kanal AWGN dengan EbNo 7 dB, sistem hybrid DS/FH spread spectrum



Gambar 9. BER hybrid DS/FH spread spectrum dan 4-FSK pada kanal AWGN dengan EbNo 10 dB dan single tone jamming

memiliki bit error rate (BER) bernilai “NA”. BER dengan nilai “NA” disini berarti dalam pengambilan data maksimum sebanyak 100000 data untuk masing-masing data informasi yang dikirimkan dan data yang diterima dan kemudian dibandingkan terindikasi tidak terdapat kesalahan atau error untuk data yang diterima. Dan dalam hal ini, keadaan tersebut akan disebut sebagai nilai BER minimum. Sedangkan untuk sistem komunikasi dengan modulasi 4-FSK memiliki nilai BER minimum pada kanal AWGN dengan EbNo 10 dB. Nilai BER maksimum pada kedua sistem terjadi pada kanal AWGN dengan EbNo 0 dB dengan BER untuk hybrid DS/FH spread spectrum bernilai 0.044642857 dan BER untuk sistem komunikasi dengan modulasi 4-FSK bernilai 0.181488203. Dari data penjelasan diatas diambil kesimpulan bahwa sistem hybrid DS/FH spread spectrum memiliki kehandalan terhadap kondisi kanal AWGN yang lebih baik daripada sistem komunikasi dengan modulasi 4-FSK.

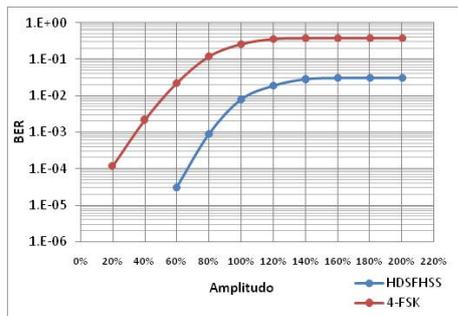
##### B. Pengaruh Single tone Jamming

Pengujian dilakukan pada kanal AWGN dengan EbNo 10 dB dengan asumsi bahwa pada EbNo 10 dB kedua sistem yang akan diuji memiliki nilai BER minimum. Parameter yang dirubah-rubah atau divariasikan adalah daya atau amplitudo dari single tone jamming dan frekuensi tone jamming. Daya atau amplitudo single tone jamming yang digunakan mulai dari 20% sampai dengan 200% yang dihitung dari daya pada frekuensi carrier.

Dari grafik pada gambar 9 dapat terlihat bahwa semakin besar daya (amplitudo) single tone jamming match semakin besar pula nilai BER-nya. Single tone jamming match tidak terlalu berpengaruh pada sistem komunikasi hybrid DS/FH spread spectrum terbukti dengan nilai BER maksimal bernilai 0.00001 yang berarti hanya ada satu data yang salah dari 100000 data yang diambil. Berbeda dengan sistem komunikasi hybrid DS/FH spread spectrum, single tone jamming match sangat berpengaruh terhadap sistem komunikasi dengan modulasi 4-FSK terbukti dengan nilai BER maksimum bernilai 0.490196 yang berarti ada hamper setengah dari data yang diambil terjadi kesalahan.

##### C. Pengaruh Multitone Jamming

Seperti pada pengujian pengaruh jamming berupa single tone jamming, pengujian dilakukan pada kanal AWGN dengan EbNo 10 dB dengan asumsi bahwa pada EbNo 10 dB kedua sistem yang akan diuji memiliki nilai BER minimum. Parameter yang dirubah-rubah atau divariasikan adalah daya atau amplitudo dari multitone jamming dan frekuensi tone jamming. Daya atau amplitudo multitone jamming yang digunakan mulai dari 20% sampai



Gambar 10. BER hybrid DS/FH spread spectrum dan 4-FSK pada kanal AWGN dengan EbNo 10 dB dengan multitone jamming

dengan 200% yang dihitung dari daya pada sinyal carrier sistem komunikasi.

Berdasarkan grafik pada Gambar 10, sistem komunikasi hybrid DS/FH spread spectrum memiliki nilai BER maksimum 0.03124 jauh lebih kecil jika dibandingkan dengan sistem komunikasi dengan modulasi 4-FSK yang memiliki nilai BER maksimum 0.37879. Untuk kedua sistem tersebut diatas terjadi kenaikan BER yang cukup signifikan pada amplitudo atau daya yang digunakan pada multitone jamming antara 60% sampai 140%, sedangkan untuk nilai amplitudo pada multitone jamming antara 160% sampai 200% nilai BER pada kedua sistem cenderung konstan. Semakin besar daya atau amplitudo pada multitone jamming, nilai BER pada sistem komunikasi yang dipengaruhi akan semakin besar.

## V. KESIMPULAN/RINGKASAN

Dari segala yang dilakukan dalam tugas akhir tentang implementasi hybrid DS/FH spread spectrum menggunakan DSK TMS320C6416T ini, meliputi tahap perencanaan blok sistem, simulasi sistem, dan implementasi sistem, terdapat beberapa kesimpulan yang dapat diambil sebagai berikut.

1. Hybrid DS/FH spread spectrum merupakan sistem komunikasi yang handal pada kondisi kanal yang buruk. Hal tersebut dibuktikan dengan kinerja sistem pada kanal AWGN dengan EbNo 0 memiliki bit error rate (BER) 0.044642857.
2. Hybrid DS/FH spread spectrum juga merupakan sistem komunikasi yang tahan terhadap adanya jamming yang berupa singletone jamming dan multitone jamming. Hal tersebut dibuktikan dengan kinerja sistem pada kanal AWGN dengan EbNo 10 memiliki BER 0.00001 untuk penambahan singletone jamming dan 0.03124 untuk penambahan multitone jamming dengan parameter amplitudo tone jamming (single dan multi) 2 kali amplitudo frequency carrier yang digunakan.
3. Kelemahan dari sistem komunikasi hybrid DS/FH spread spectrum ini yaitu terlalu boros dalam penggunaan spektrum frekuensi. Hal tersebut dibuktikan dengan mengirim data informasi dengan laju data 1 data/detik, sistem ini membutuhkan spektrum frekuensi dari 4 Hz sampai 64 Hz.

## DAFTAR PUSTAKA

- [1] Sklar, Bernard. "Digital Communications Fundamentals and Applications". Prentice Hall. California. 2001.
- [2] Torrieri, Don. "Principles of Spread-Spectrum Communication Systems 2<sup>nd</sup> Edition". Springer. New York. 2011.
- [3] Hayes, Monson H. "Schaum's Outline of Theory and Problems of Digital Signal Processing". The McGraw-Hill Companies, Inc. United States of America. 1999.
- [4] Chassaing, Rulph. "Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK". JOHN WILEY & SONS, INC. USA. Second Edition, 2008.
- [5] \_\_\_\_\_, "TMS320C6416T DSK Technical Reference". SPECTRUM DIGITAL, INC. 2004.
- [6] \_\_\_\_\_, "Praktikum Pengolahan Sinyal – Code Composer Studio Basic". Laboratorium Pengolahan Sinyal E-206 (D4 Lintas Jalur) PENS-ITS. Surabaya. 2006.
- [7] Mathworks. "Documentation". <http://www.mathworks.com/help/>