

# Implementasi dan Evaluasi Kinerja *Frequency Hopping Spread Spectrum* (FHSS) Menggunakan WARP

Rini Nurpita Sari, Suwadi, dan Titiek Suryani

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

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: rininurpitasari@gmail.com, suwadi110@gmail.com, titiks@ee.its.ac.id

**Abstrak**— Proses pengiriman informasi dari pengirim ke penerima terkadang terdapat gangguan, maka untuk mengatasi masalah tersebut dapat menggunakan metode *Spread Spectrum*, yang merupakan teknik pengiriman sinyal yang tahan terhadap gangguan berupa *noise* maupun *jamming*. *Frequency Hopping Spread Spectrum* merupakan sinyal informasi ditransmisikan yang dilompat ke dalam suatu spektrum frekuensi secara random atau sesuai pola tertentu. Spektrum tersebut sudah dialokasikan menjadi beberapa *channel* yang diperuntukkan untuk *frequency hopping signal*. Dalam Tugas Akhir ini, akan dilakukan implementasi sistem komunikasi SISO pada FHSS menggunakan *Wireless Open Access Research Platform* (WARP). Sistem FHSS ini akan dimodelkan dengan menggunakan *software* Matlab. Ketahanan sistem FHSS terhadap *Single Tone Jamming* dan *Multi Tone Jamming* akan diketahui dengan melakukan beberapa pengujian. Hasil yang diperoleh nilai *Bit Error Rate* (BER) akan semakin kecil jika daya yang dipancarkan semakin besar, sedangkan pada *jamming* dipengaruhi oleh nilai daya *jamming*. Penelitian ini menunjukkan bahwa kinerja sistem berjalan baik saat *Single Tone Jamming* lebih baik dari pada *Multi Tone Jamming*. Hal ini dibuktikan dengan hasil nilai BER saat daya *jamming* 300% untuk *Single Tone Jamming* sebesar 0.4194, dan *Multi Tone Jamming* sebesar 0.4892. Rata-rata sistem tidak terpengaruh oleh gangguan *jamming*, saat nilai daya *jamming* 75% dan 100%. Hasil penelitian ini menunjukkan bahwa sistem FHSS lebih tahan terhadap *Single Tone Jamming* dan *Multi Tone Jamming* dibandingkan sistem 4-ary FSK.

**Kata kunci**— *Bit Error Rate*, *Frequency Hopping*, *Jamming*, *Spread Spectrum*, WARP

## I. PENDAHULUAN

PADA saat proses pengiriman informasi sering dijumpai masalah interferensi, *noise*, maupun *jamming* yang bisa merusak sinyal informasi. Untuk mengatasi masalah tersebut maka digunakan teknik komunikasi *frequency-hopping spread spectrum*. Teknik tersebut digunakan karena sinyal pembawa (*carrier*) mengubah-ubah frekuensi (*frequency-hopping*) secara acak tetapi sesuai urutan. Jadi dapat mengurangi kerusakan yang ditimbulkan oleh interferensi, *noise*, maupun *jamming* dan menjamin kerahasiaan sinyal yang dikirimkan.

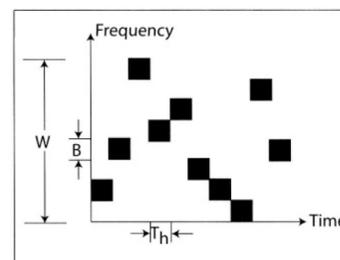
Pada proyek akhir ini akan melakukan simulasi pengiriman dan penerimaan sinyal informasi dari pemancar ke penerima dengan teknik *frequency-hopping spread spectrum* disertai gangguan (*jamming*) yaitu *Single* dan *Multi Tone Jamming*. Simulasi dilakukan dengan bantuan Matlab, kemudian diimplementasikan pada *Wireless Open Access Research Platform* (WARP) adalah salah satu dari perangkat yang bisa digunakan untuk memvisualisasikan

proses pengiriman dan penerimaan sinyal informasi dan didukung perangkat lainnya.

## II. TEORI PENUNJANG

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

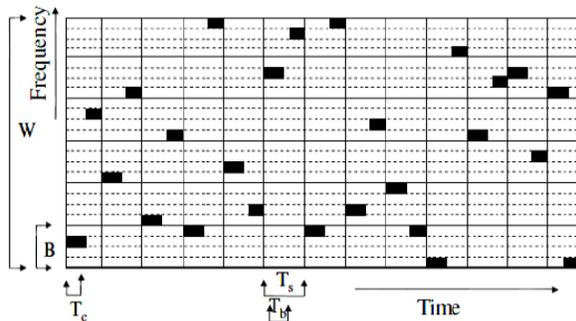
*Frequency Hopping Spread Spectrum* merupakan sinyal informasi ditransmisikan yang disebar secara random ke beberapa frekuensi radio. Sinyal informasi tersebut dilompatkan ke dalam suatu spektrum frekuensi, dimana spektrum tersebut sudah dialokasikan menjadi beberapa *channel* yang diperuntukkan untuk *frequency hopping signal*. Lompatan sinyal informasi tersebut dapat dilakukan secara random ataupun menurut aturan tertentu. Formasi dari lompatan tersebut diatur berdasarkan kode tertentu, yang dinamakan chipping code, artinya termasuk bagian dari encrypting signal. Suatu spektrum atau rentang frekuensi akan disekat atau dibagi menjadi *channel-channel* teralokasi yang akan digunakan untuk melompatkan sinyal informasi dalam FHSS. Pembagian *channel-channel* tersebut didasarkan atas interval yang besarnya tetap, sehingga sinyal informasi akan dilompatkan dari satu frekuensi ke frekuensi lainnya dengan interval frekuensi yang tetap. Jarak interval tersebut dipengaruhi lebar dari spektrum yang digunakan untuk FHSS dan juga dipengaruhi oleh besarnya suatu sinyal informasi. Lebar dari beberapa *channel* yang digunakan sebagai lompatan dalam FHSS juga akan tergantung dari besarnya bandwidth dan input sinyal informasi. Apabila input sinyal informasi mempunyai bandwidth yang besar, maka otomatis lebar dari masing-masing *channel* sebagai tempat lompatannya juga akan besar, begitu pula sebaliknya. Gambar 1 ilustrasi kanal frekuensi dikaitkan dengan keterangan *frequency-hopping pattern*. Interval waktu antara hop disebut *hop interval*. Durasinya disebut *hop duration* dan dilambangkan dengan  $T_h$  [1].



Gambar 1 Pola *Frequency-hopping* [1]

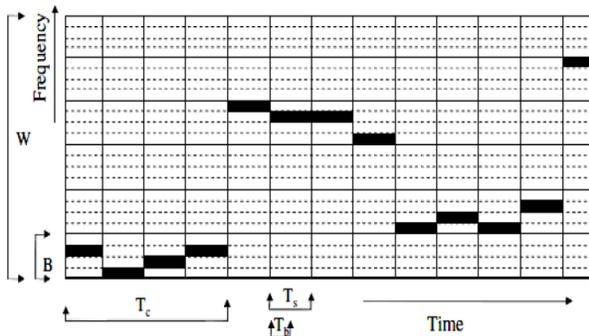
Frekuensi *hopping* dapat diklasifikasikan sebagai *fast* dan *slow*. *Fast frequency hopping* (FFH) terjadi jika ada lebih dari satu hop untuk setiap simbol informasi. Meskipun

definisi ini tidak mengacu pada *hop rate*, *Fast frequency hopping* (FFH) merupakan pilihan hanya jika *hop rate* yang melebihi *information-symbol rate* dapat diimplementasikan. Pada *Fast frequency hopping* (FFH) nilai  $T_c < T_s$  yaitu frekuensi hopping terjadi lebih cepat dari modulasi. Sistem *fast FH/MFSK* berbeda dari sistem *slow FH/MFSK* bahwa ada beberapa lompatan setiap simbol M-ary. Oleh karena itu, dapat diterapkan dari *Fast frequency hopping* (FFH) yaitu  $T_s = NT_c$ . Gambaran dari sistem *fast FHSS* dengan modulasi 4-FSK ( $M = 4$ ), 6 frekuensi lompatan ( $L = 6$ ), dan 2 lompatan setiap simbol ( $T_s = 2T_c$ ,  $N = 2$ ) [2].



Gambar 2 *Frequency Plot* dari *Fast frequency hopping* [2]

*Slow frequency hopping* (SFH) terjadi jika salah satu atau lebih simbol informasi yang ditransmisikan dalam interval waktu antara hop frekuensi. *Slow frequency hopping* (SFH) biasanya lebih baik karena gelombang yang ditransmisikan jauh lebih rapat dan *switching time* yang berlebihan bisa berkurang. Pada *Slow frequency hopping* (SFH) ini dimana  $T_c > T_s$ . Sinyal *slow FH/MFSK* adalah karakteristik dengan memiliki beberapa simbol transmisi setiap hop. Oleh karena itu, setiap simbol dari sinyal *slow FH/MFSK* adalah sebuah chip. Ketentuan dari *Slow frequency hopping* (SFH) adalah  $T_c = NT_s$ . Persyaratan orthogonalitas untuk sinyal FSK pemisahan antara simbol frekuensi FSK yang berdekatan setidaknya  $2\pi/T_s$ . Oleh karena itu, pemisahan minimum antara frekuensi hopping berdekatan adalah  $2\pi M/T_s$ . *Slow frequency hopping* (SFH) dengan 4-FSK ( $M = 4$ ), 6 frekuensi hop ( $L = 6$ ), dan 4 simbol setiap lompatan ( $t_c = 4T_s$ ,  $N = 4$ ) [2].



Gambar 3 *Frequency Plot* dari *Slow frequency hopping* [2]

- Dimana :  $T_c$  = Hop period
- $T_s$  = Symbol period
- $T_b$  = Bit period
- $W$  = Spread bandwidth
- $B$  = Symbol bandwidth

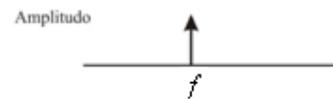
### B. Modulasi MFSK

Pada sistem frekuensi hopping ini menggunakan *Multiple Frequency Shift Keying* (MFSK) sebagai modulasi datanya. Frekuensi pembawa yang dipilih dari salah satu frekuensi  $q$  untuk setiap simbol yang ditransmisikan, dan kumpulan dari frekuensi tersebut memungkinkan perubahan frekuensi di setiap hop. Sistem FH/MFSK dimana keluaran bit dari generator kode dan input digital digabungkan untuk menentukan frekuensi yang dihasilkan oleh *synthesizer*. MFSK merupakan fase kontinu dari simbol ke simbol, jika tidak dapat dimodelkan sebagai variabel acak terdistribusi merata secara menyeluruh. Implementasi dari kontinuitas fase sangat dibutuhkan untuk mencegah hamburan spektral yang berlebihan di luar kanal frekuensi [1].

Pada implementasi standar, sub kanal  $q$  dari setiap kumpulan MFSK berdekatan, dan setiap kumpulan merupakan kanal frekuensi dalam *hopping band*. Untuk sinyal orthogonal non-koheren, pola MFSK harus benar-benar dipisahkan agar sinyal yang diterima menghasilkan respon yang dapat diabaikan di sub-kanal yang salah. Seperti yang ditunjukkan selanjutnya, pemisahan frekuensi harus  $f_d = k/T_s$ , di mana  $k$  adalah bilangan bulat *nonzero*, dan  $T_s$  menyatakan durasi simbol.

### C. Single Tone Jamming

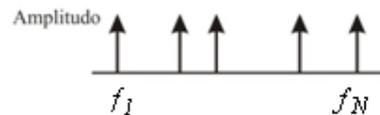
Sebuah sinyal *jamming* merupakan sebuah pola *Continuous Wave* (CW) yang diletakkan di frekuensi tunggal. Dengan demikian, *Single Tone Jamming* (STJ) merupakan sebuah frekuensi tunggal yang terletak dalam *bandwidth* yang ditentukan dari sinyal yang macet, ditunjukkan pada gambar 4. *Single tone jamming* juga disebut dengan *spot jamming* [3].



Gambar 4 *Single Tone Jamming* [3]

### D. Multiple Tone Jamming

*Multiple Tone Jamming* (MTJ) merupakan *jamming* yang dapat mengganggu sinyal saluran beberapa atau seluruh penerima. *Multiple-tone jamming* mendistribusikan daya *jammer* di antara beberapa pola (*tones*) dan ditunjukkan pada gambar 5.



Gambar 5 *Multiple Tone Jamming* [3]

*Jammer* akan memancarkan lebih dari satu pola (*tone*) yang ditempatkan secara acak atau ditempatkan pada frekuensi tertentu. Jika target khusus sistem komunikasi AJ yang rentan terhadap pola tertentu dan *jammer* mengetahuinya, maka mungkin lebih bijaksana untuk menggunakan pola pada frekuensi tertentu daripada menempatkannya secara acak.

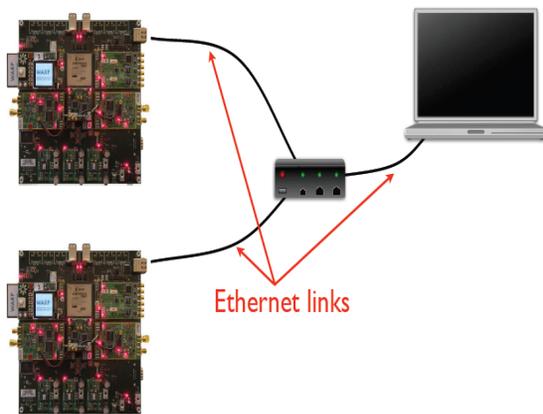
### E. WARP (Wireless Open-Access Research Platform)

WARP (*Wireless Open-Access Research Platform*) merupakan bentuk desain *platform wireless* yang dapat di program untuk membuat *prototype* jaringan *wireless*,

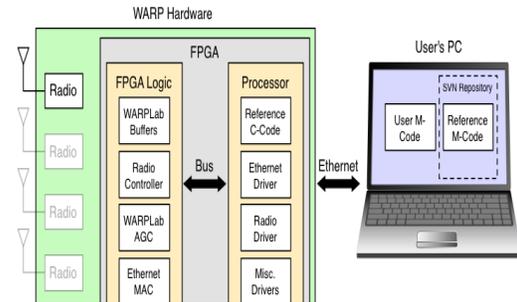
sebagai *prototype* komunikasi *wireless* untuk penelitian dalam algoritma nirkabel yang maju saat ini. WARP menggabungkan perangkat dengan kemampuan tinggi yang bisa diprogram dengan *repository open-source*. WARP merupakan salah satu bentuk *Software Defined Radio* yang dikembangkan sebagai modul penelitian tentang sistem komunikasi nirkabel. WARP sebuah *Field Programmable Gate Array* (FPGA) atau modul *Digital Signal Processing* (DSP) yang terhubung dengan modul radio. Pada WARP memiliki Software WARPLab yang berkerja melalui Matlab untuk memberikan perintah pada WARP. Tujuan *platform* dapat digambarkan pada level tinggi karena memiliki empat syarat. Pertama, *platform* mampu mengimplementasikan algoritma nirkabel yang canggih, memungkinkan pengolahan sumber daya tambahan untuk dialokasikan ketika komputasi daya dari proses tunggal membuktikan tidak cukup, memberikan ekspansi *interface* untuk mengakomodasi aplikasi masa depan, dan untuk melestarikan utilitas dalam jangka panjang yang *platform* harus mendukung evolusi beberapa komponen tanpa merusak fungsi lain [4].

WARPLab merupakan kerangka kerja yang digunakan untuk desain penggabungan WARP dan Matlab. Penggunaan dari WARPLab memungkinkan kombinasi dari beberapa sistem multi-antena pada pemancar dan penerima. WARPLab berinteraksi dengan node WARP langsung dari *workspace* Matlab dan sinyal yang dihasilkan dalam Matlab dapat ditransmisikan secara *real-time* melalui udara menggunakan node WARP, untuk gambaran dari WARPLab ditunjukkan pada gambar 6 [5].

Masing-masing *node* yang diprogram melalui WARPLab terhubung dengan komputer melalui switch Ethernet. Dengan menggunakan sistem ini, *node* dapat diprogram untuk mengirim data dan menerima data melalui MATLAB. *Setup* WARPLab ditunjukkan pada gambar 7.



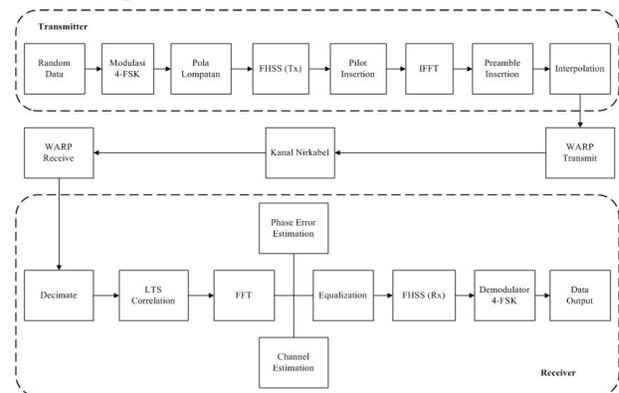
Gambar 6 Gambar WARPLab [5]



Gambar 7 Setup WARPLab [6]

### III. PERANCANGAN DAN IMPLEMENTASI SISTEM

Sebagai acuan dari desain sistem komunikasi *frequency hopping spread spectrum* (FHSS) akan digunakan program yang dibuat oleh Universitas Rice. Pada program ini, dengan menggunakan WARPLab pada Matlab, sistem komunikasi antara dua modul WARP bisa dilakukan. Pada penerapan program Matlab yang sudah di desain akan diterapkan pada WARPLab versi 7. Secara garis besar, proses sistem komunikasi FHSS dilakukan dengan langkah-langkah yang ditunjukkan pada gambar 8.



Gambar 8 Pemodelan Sistem Komunikasi dengan Teknik FHSS

Perancangan dan implementasi sistem dibuat blok diagram yang berpedoman pada gambar 8. Dalam blok diagram Tugas Akhir yang dirancang, sinyal masukan akan dibangkitkan secara acak atau *random* dengan menggunakan perintah *randint* pada Matlab dengan *state* tertentu sehingga hasilnya akan saling bebas dan tidak mempengaruhi satu dengan yang lain serta memungkinkan untuk pengulangan atau mendapatkan data yang sama pada *state* yang sama.

Bit yang dibangkitkan secara acak berupa bernilai '1' dan '0' ini, selanjutnya bit-bit tersebut sebanyak *n* menjadi sinyal yang bernilai *integer* dengan 0 hingga  $(2^n)-1$  sesuai jenis modulasi yang telah ditentukan MFSK ( $m = 2^n$ , dimana  $n = 2, 4, 6$ ). Pada Tugas Akhir ini digunakan  $n = 2$ , sehingga ada 4 kemungkinan nilai simbolnya yaitu [00 01 10 11]. Simbol-simbol tersebut kemudian dikonversi menjadi nilai *integer* (0-3).

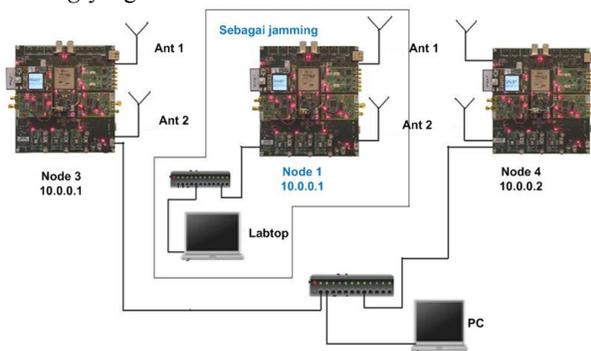
Setelah itu, akan di bentuk pola lompatan, pola lompatan adalah nilainya dari 0 sampai 11, pola lompatan ini terus berulang selama sistem berjalan. Pola lompatan yang digunakan pada Tugas Akhir ini adalah '5 9 1 6 2 7 11 4 8 3 10 0'. Kemudian akan dibentuk parallel dengan ukuran 48x120 untuk penentuan data FHSS. Selanjutnya masuk ke dalam proses IFFT. Dalam proses IFFT ini memanfaatkan

modulasi OFDM untuk pengimplementasian sistem komunikasi SISO pada FHSS dengan menggunakan WARP. Proses IFFT pada Tugas Akhir ini dibentuk dari matriks berukuran 64x120. Dalam matriks berukuran 64x120 tersebut terdapat *subcarrier* pilot yang terletak pada baris (8, 22, 44, 58), sedangkan untuk *subcarrier* data terdapat pada baris (2:7 9:21 23:27 39:43 45:57 59:64).

Proses penambahan *preamble*, yang berfungsi untuk digunakan mendeteksi data awal pada penerima. *Preamble* ini berupa angka scalar. Penelitian yang dilakukan pada FHSS ini menggunakan *preamble* sepanjang 320, payload sepanjang 7680 dan zero padding sepanjang 192. Setelah proses IFFT ini, selanjutnya proses interpolasi yaitu proses sampling rate untuk sinyal pada frekuensi yang lebih tinggi agar mencapai *buffer* 2<sup>14</sup>. Setelah itu proses dengan WARP, yaitu transmisi sinyal melalui *Ethernet Switch* menggunakan *node* WARP. Implementasi ke WARP ini menggunakan fungsi Matlab yang sudah disediakan oleh WARPLab.

Selanjutnya, proses di penerima yaitu proses *decimate* yang merupakan kebalikan dari proses interpolasi. Pada proses *decimate* ini nilai matriks *interpolate* 1x16384 akan menjadi 1x8192. Proses selanjutnya adalah LTS *Correlation*, merupakan proses *cross correlation* antara *preamble* yang terdapat pada penerima dan satu LTS yang terdapat pada pemancar. Selanjutnya proses FFT dengan ukuran matriks yang dihasilkan FFT harus sama dengan IFFT yaitu 64x120. Setelah proses FFT langkah selanjutnya yaitu proses estimasi kanal untuk domain frekuensi LTS. Selanjutnya proses phase error estimation untuk sisa dari domain frekuensi pilot tones. Setelah proses estimasi kanal dan phase error estimation, kemudian proses equalizer yang merupakan untuk mengatasi nilai magnitudo dan respon fasa yang rusak. Mengatasi nilai magnitudo menggunakan hasil dari *cross-corelation* sedangkan untuk mengatasi respon fasa menggunakan pilot. Selanjutnya masuk ke proses demodulasi yang merupakan kebalikan dari fungsi modulasi, proses demodulasi selesai maka didapat keluaran bit informasi yang memiliki jumlah yang sama antara bit informasi yang di bangkitkan pada pemancar.

Pada sistem FHSS yang di implementasikan tanpa *jamming*, dilakukan juga dengan menggunakan gangguan *jamming*, data dikirim melalui dua *Node* WARP, tetapi untuk sistem FHSS dengan *jamming* membutuhkan satu *node* WARP lagi sebagai *jamming*nya. Integrasi sistem FHSS dengan *jamming* menggunakan *node* 1 sebagai *jamming* yang diletakan diantara *node* 3 dan *node* 4.



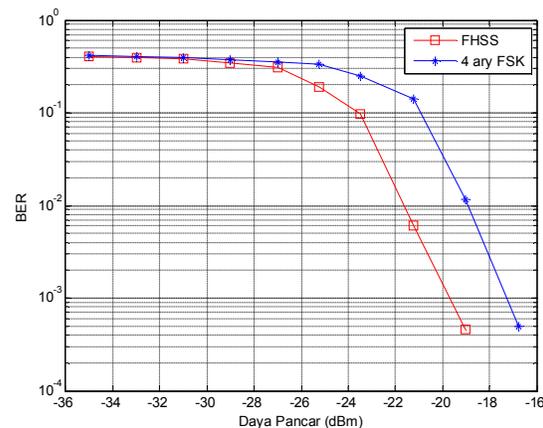
Gambar 9 Konfigurasi Sistem FHSS dengan *Jamming*

#### IV. PENGUJIAN DAN ANALISA DATA

Dalam bab ini akan dibahas mengenai pengujian implementasi sistem komunikasi SISO dengan teknik *Frequency Hopping Spread Spectrum* (FHSS) menggunakan *Wireless Open Access Research Platform* (WARP). Hasil pengujian kinerja sistem FHSS secara implemetasi dengan disajikan dalam bentuk grafik BER vs Daya Pancar (dBm), sedangkan untuk kinerja sistem FHSS terhadap *Single Tone Jamming* dan *Multi Tone Jamming* dalam bentuk grafik BER vs Daya *Jamming* (%). Terdapat beberapa model sistem FHSS yang dapat dianalisa dari Tugas Akhir ini yaitu Kinerja sistem FHSS tanpa *jamming*, Kinerja sistem FHSS terhadap gangguan *Single Tone Jamming* (STJ), dan Kinerja sistem FHSS terhadap gangguan *Multi Tone Jamming* (MTJ). Selain dilakukan pengujian terhadap tiga model sistem FHSS diatas, juga akan dilakukan pengujian sistem 4-ary FSK sebagai pembanding. Pada implementasi sistem *Frequency Hopping Spread Spectrum* (FHSS) ini, dilakukan 10 kali pengujian. Kondisi lingkungan pengukuran dilakukan di ruang B.304 Laboratorium Telekomunikasi Multimedia Jurusan Teknik Elektro. Pengujian dilakukan diruang B.304 dengan jarak 6 meter. Jarak antara *node* pengirim (WARP *node* 3) dan penerima (WARP *node* 4) berjarak sejauh 6 mater.

##### A. Analisa Kinerja Implementasi sistem tanpa *Jamming*

Pengukuran sistem *Frequency Hopping Spread Spectrum* (FHSS) pada implementasi modul WARP menggunakan parameter pengukuran dengan nilai Tx RF = 0:4:60, Tx BB = 1, Rx RF = 1, dan Rx BB = 1. Untuk hasil pengujian pada modul WARP ini pada sistem FHSS akan disajikan dalam bentuk grafik BER vs Daya Pancar (dBm), dan dibandingkan dengan 4-ary FSK.



Gambar 10 Grafik BER vs Daya Pancar (dBm) untuk Sistem FHSS dan 4-ary FSK tanpa *Jamming*

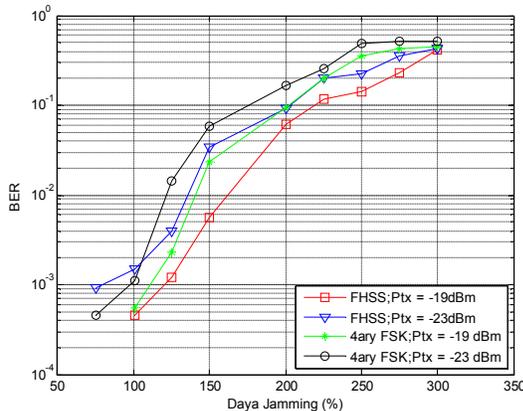
Berdasarkan gambar 10, grafik BER vs daya pancar terhadap pengukuran hasil tanpa *jamming* menunjukkan bahwa semakin besar daya yang dipancarkan maka semakin kecil nilai BER yang dihasilkan. Hal ini dipengaruhi oleh daya yang dipancarkan pada sisi pengirim, semakin besar daya yang dipancarkan maka pengaruh *noise* yang ada untuk mempengaruhi informasi dalam pengiriman data juga akan semakin kecil, serta daya yang diterima di penerima semakin besar pula, sehingga mengakibatkan nilai BER yang dihasilkan pada sistem FHSS dan 4-ary FSK akan

semakin kecil. Jadi dapat disimpulkan dari hasil pengukuran implementasi pada modul WARP terhadap sistem FHSS dan 4-ary FSK, dimana nilai BER berbanding terbalik dengan daya pancar.

Selain pengujian tanpa *jamming*, juga dilakukan pengujian menggunakan *jamming*. Dalam pengujian sistem ini akan diuji dengan dua model gangguan *jamming*, yaitu *Single Tone Jamming* (STJ) dan *Multi Tone Jamming* pada nilai daya pancar (Ptx) -19 dBm dan -23 dBm. Pada implementasi ini, nilai BER yang didapat dari masing-masing model akan dibandingkan terhadap perubahan daya sinyal *jamming*. Setiap modelnya akan dipengaruhi sinyal *jamming* dengan frekuensi berbeda agar terlihat pengaruh sistem yang didesain. Pada pengujian sistem dengan *Single Tone Jamming* dan *Multi Tone Jamming*, parameter yang digunakan adalah perubahan frekuensi sinyal *jamming* dan daya *jamming* terhadap perubahan nilai *bit error rate*.

**B. Frekuensi jamming pada Single Tone Jamming sama dengan frekuensi jamming**

Pada pengujian sistem FHSS dengan gangguan *Single Tone Jamming* akan diuji dengan menggunakan frekuensi *jamming* sama dengan frekuensi kerja yaitu 2.432 GHz, serta dengan menggunakan nilai Ptx yang sama antara sistem FHSS dan 4-ary FSK.



Gambar 11 Grafik BER vs Daya Sinyal *Jamming* pada Frekuensi kerja sama dengan Frekuensi *jamming* saat *Single Tone Jamming*

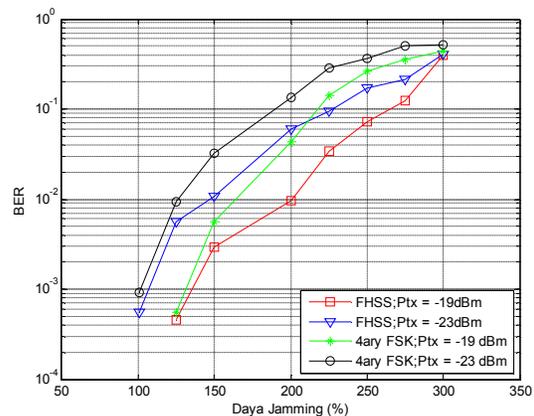
Berdasarkan gambar 11 nilai BER yang dihasilkan saat menggunakan frekuensi *jamming* sama dengan frekuensi kerja mengalami jumlah *Bit Error Rate* yang besar. Dilihat berdasarkan daya pancar, bahwa disaat nilai daya pancar yang digunakan besar yaitu -19 dBm maka akan menghasilkan nilai BER yang kecil dan sebaliknya untuk nilai daya pancar yang kecil. Pada hasil pengujian frekuensi *jamming* sama dengan frekuensi kerja ini, nilai BER yang didapat akan semakin besar jika nilai daya *jamming* yang digunakan semakin besar juga. Pada sistem FHSS dan 4-ary FSK tidak terpengaruh oleh *jamming* saat daya *jamming* 75% sampai 10% untuk nilai Ptx = -19 dBm dan untuk Ptx = -23 dBm pada 50% sampai 10%.

**C. Frekuensi sinyal jamming tidak sama dengan frekuensi kerja pada pengujian Single Tone Jamming**

Pada pengujian sistem FHSS dengan gangguan *Single Tone Jamming* saat menggunakan frekuensi *jamming* tidak

sama dengan frekuensi kerja, frekuensi yang digunakan adalah 2.433 GHz.

Berdasarkan gambar 12, menghasilkan nilai BER yang lebih kecil dari pada nilai BER terhadap pengujian menggunakan frekuensi *jamming* sama dengan frekuensi kerja. Saat nilai daya *jamming* 300 % menghasilkan nilai BER untuk sistem FHSS pada Ptx = -19 dBm sebesar 0.3927 dan saat Ptx = -23 dBm menghasilkan nilai BER sebesar 0.4021, sedangkan pada hasil pengujian frekuensi *jamming* sama dengan frekuensi kerja nilai BER untuk sistem FHSS saat Ptx = -19 dBm sebesar 0.4194 dan Ptx = -23 dBm sebesar 0.4235. Daya *jamming*, semakin besar nilai daya *jamming* yang dilakukan maka akan semakin besar juga nilai BER yang dihasilkan. Pada hasil pengujian terhadap frekuensi *jamming* tidak sama dengan frekuensi kerja, *Single Tone Jamming* tidak terpengaruh pada Ptx = -19 dBm dengan daya *jamming* 100% sampai 10% sedangkan pada Ptx = -23 dBm saat daya *jamming* 75% sampai 10%.



Gambar 12 Grafik BER vs Daya Sinyal *Jamming* pada Frekuensi kerja tidak sama dengan Frekuensi *jamming* saat *Single Tone Jamming*

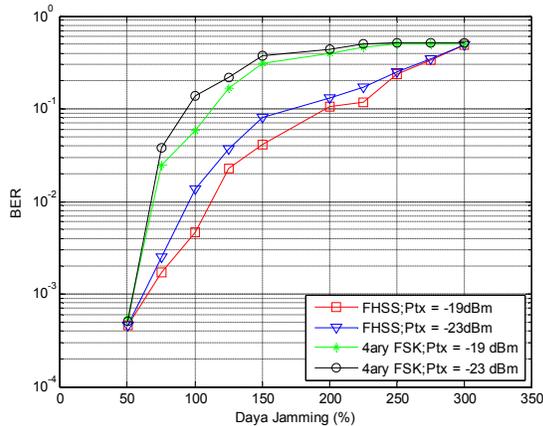
**D. Frekuensi sinyal jamming sama dengan frekuensi kerja pada pengujian Multi Tone Jamming**

Frekuensi *jamming* disesuaikan dengan salah satu dari frekuensi kerja, tetapi untuk pengujian kali ini frekuensi *jamming* satunya dibawah frekuensi kerja. Frekuensi *jamming* yang pertama menggunakan frekuensi pada 2.431 GHz dan frekuensi *jamming* kedua menggunakan fekuensi pada 2.432 GHz.

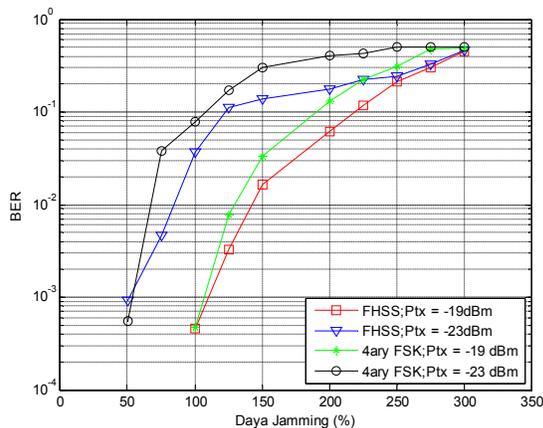
Gambar 13 menghasilkan nilai BER yang lebih besar dibandingkan dengan hasil pengujian *Single Tone Jamming* saat menggunakan frekuensi *jamming* sama dengan frekuensi kerja, yaitu misalnya pada sistem FHSS saat Ptx = -19 dBm dan daya *jamming* sebesar 300% menghasilkan nilai BER sebesar 0.4892 sedangkan untuk *Single Tone Jamming* menghasilkan nilai BER sebesar 0.4194. Hal ini karena *tone* yang digunakan sebanyak dua buah dan juga pengaruh frekuensi yang digunakan. Berdasarkan grafik pada gambar 13 dapat disimpulkan bahwa semakin besar nilai daya *jamming* maka nilai bit yang salah akan semakin besar. Nilai kesalahan bit sistem FHSS lebih kecil dari sistem 4-ary FSK. Pengujian pada *Multi Tone Jamming* ini, sistem FHSS dan 4-ary FSK pada daya *jamming* sebesar 50% ke bawah tidak dipengaruhi oleh *jamming*. Berdasarkan hasil pengujian pada *Multi Tone Jamming* ini, sistem FHSS lebih baik dari pada sistem 4-ary FSK.

E. Frekuensi sinyal jamming tidak sama dengan frekuensi kerja pada Multi Tone Jamming

Pengujian ini dilakukan pada Multi Tone Jamming dengan frekuensi jamming tidak sama dengan frekuensi kerja. Kedua frekuensi tidak ada yang sama dengan frekuensi kerja, frekuensi jamming yang pertama menggunakan frekuensi pada 2.431 GHz dan frekuensi jamming kedua menggunakan fekuensi pada 2.433 GHz.



Gambar 13 Grafik BER vs Daya Sinyal Jamming pada Frekuensi kerja sama dengan Frekuensi jamming saat Multi Tone Jamming



Gambar 14 Grafik BER vs Daya Sinyal Jamming pada Frekuensi kerja tidak sama dengan Frekuensi jamming saat Multi Tone Jamming

Berdasarkan Gambar 14 menghasilkan nilai BER yang lebih kecil dari pada nilai BER pada gambar 13. Hasil pengujian terhadap frekuensi jamming tidak sama dengan frekuensi kerja, pada sistem FHSS dan 4-ary FSK untuk nilai Ptx = -19 dBm tidak dipengaruhi oleh gangguan Multi Tone Jamming saat daya jamming sebesar 75% sampai dengan 10%. Sedangkan untuk Ptx = -23 dBm tidak dipengaruhi oleh gangguan Multi Tone Jamming pada 50% sampai 10%.

V. KESIMPULAN/RINGKASAN

Hasil implementasi pengujian untuk sistem FHSS dan 4-ary FSK tanpa jamming, dipengaruhi oleh nilai Tx Rf jika semakin besar Tx Rf yang dipancarkan maka akan menghasilkan nilai BER yang semakin kecil. Pada menggunakan jamming berpengaruh terhadap nilai daya jamming yang digunakan, semakin besar daya jamming maka akan semakin besar juga nilai BER yang dihasilkan.

Misalnya pada Single Tone Jamming saat daya jamming 300% menghasilkan nilai BER sebesar 0.4194 dan saat daya jamming 100% menghasilkan nilai BER sebesar 0.00046296. Single Tone Jamming lebih baik dari pada Multi Tone Jamming, pada Single Tone Jamming menghasilkan nilai BER saat daya jamming 300% sebesar 0.4194 sedangkan pada Multi Tone Jamming saat daya jamming 300% juga menghasilkan nilai BER sebesar 0.4892.

Gangguan Single Tone Jamming untuk frekuensi jamming sama dengan frekuensi kerja tidak terpengaruh oleh jamming saat kondisi Ptx = -19 dBm dengan daya jamming 75% dan Ptx = -23 dBm pada daya jamming 50%, sedangkan untuk frekuensi jamming tidak sama pada Ptx = -19 dBm saat daya jamming 100% dan Ptx = -23 dBm dengan daya jamming 75%. Pada Multi Tone Jamming untuk frekuensi jamming sama dengan frekuensi kerja tidak di pengaruhi oleh jamming pada nilai daya jamming 30% untuk Ptx = -19 dBm dan Ptx = -23 dBm, sedangkan pada frekuensi jamming tidak sama pada Ptx = -19 dBm dengan daya jamming 75% dan Ptx = -23 dBm dengan daya jamming 30%.

DAFTAR PUSTAKA

- [1] Torrieri, Don, "Principles of Spread-Spectrum Communication Systems", Springer, 2005.
- [2] Olšovský, Peter, "Design And Simulation Of Frequency Hopping Technique In Matlab", Slovak University of Technology.
- [3] Poisel, Richard, "Modern Communications Jamming Principles and Techniques", Second Edition, Artech House, 2011.
- [4] Murphy, Patrick. Ashu Sabharwal. dan Behnaam Aazhang, "Design Of WARP: A Wireless Open-Access Research Platform", 14<sup>th</sup> European Signal Processing Conference (EUSIPCO 2006), Florence, Italy.
- [5] Physical Layer Prototyping using WARPLab. [Online] November 30, 2014. [Disunting: Desember 01, 2014.] [http://warpproject.org/trac/wiki/Workshops/Rice\\_2008](http://warpproject.org/trac/wiki/Workshops/Rice_2008)
- [6] WARPLab 6 Framework Overview. WARP Project - Wireless Open Access Research Platform. [Online] September 2014. [Disunting: Desember 04, 2014.] <http://warpproject.org/trac/wiki/WARPLab6>