



Available online at : <http://bit.ly/InfoTekJar>

## InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



## Filtering Sinyal Suara Gitar Menggunakan *Band Pass Filter*

Adhif Rayana AR<sup>1</sup>, Wiwid Andriani<sup>2</sup>

<sup>1</sup> Universitas Samudra, Jl. Meurandeh, Langsa Lama, Kota Langsa, Aceh, 24354, Indonesia

### KEYWORDS

Sinyal, Suara, Gitar, *Bandpass filter*, *Fast Fourier Transform* (FFT).

### CORRESPONDENCE

Phone: +62 81210705250

E-mail: [adhifrayanaar@gmail.com](mailto:adhifrayanaar@gmail.com)

### A B S T R A C T

Pada bidang komunikasi bahwa sinyal suara asli tanpa derau (*noise*) dikatakan penting dikarenakan memberikan informasi dengan jelas, jika sinyal suara asli telah tercampur derau (*noise*) maka sinyal suara tersebut menjadi lemah dan akhirnya hilang/tidak berguna. Sehingga untuk mengatasi sinyal suara yang telah berderau (*noise*) tersebut perlu dilakukan penapisan (*filtering*) terhadap sinyal suara asli agar sinyal suara asli dapat diterima dengan baik oleh di pendengar/penerima informasi. Maka dari itu penelitian ini dilakukan untuk penapisan (*filtering*) terhadap sinyal suara asli gitar yang telah tercampur derau (*noise*) menggunakan metode *band pass filter*. Dimana metode *band pass filter* berfungsi untuk meloloskan sinyal terhadap range frekuensi diatas frekuensi batas bawah (*f<sub>L</sub>*) dan dibawah frekuensi batas atas (*f<sub>H</sub>*). Pada penelitian tersebut metode *band pass filtering* diterapkan pada file suara gitar berextensi .wav, kemudian hasil *filtering* dilakukan pembangkitan pada suara asli gitar. Dan dalam proses enkripsi, sinyal informasi yang berupa sinyal analog (domain waktu) dikonversikan kedalam domain frekuensi menggunakan algoritma *Fast Fourier Transform* (FFT). Dan penapisan (*filter*) dilakukan dengan *filter band pass*. Hasil penapisan diperoleh suatu 'Grafik' maupun 'Suara' yang mendekati Grafik dan Suara asli. Namun demikian perbandingan suara asli dan hasil *filtering* (penapisan).

## PENDAHULUAN

Dari tahun ke tahun sampai pada perkembangan di zaman saat ini tidak dapat dipungkiri lagi bahwasanya sinyal sudah merupakan bagian penting dari sebuah sistem telekomunikasi. Sehingga teknik pengolahan sinyal digital dalam sistem komunikasi perlu dipahami agar sinyal suara asli yang telah berderau dapat difiltering sampai menghasilkan sinyal suara asli kembali. Faktanya dizaman ini sudah banyak sekali aplikasi perekam yang digunakan untuk mendapatkan sebuah informasi secara instan, setelah direkam lalu ditransmisikan dan disimpan dalam format digital. Terkadang kala ketika melakukan perekaman sebuah suara tidak jarang mendapatkan suara yang berderau (*noise*) maka dari itu perlu dilakukan pengolahan sinyal digital tersebut dengan menggunakan sebuah teknik/ metode untuk mengatasi suara yang berderau (*noise*) tidak terdapat lagi *noise* dan hanya tinggal suara aslinya saja. Hasilnya, pengolahan sinyal digital menjadi suatu alat modern yang penting. Pengolahan sinyal digital antara lain: perkiraan parameter karakteristik sinyal, penghapusan atau pengurangan *noise* yang tidak diinginkan dan transformasi sinyal kedalam beberapa bentuk yang lebih informatif.

*Band pass filter* (BPF) adalah rangkaian yang melewatkan frekuensi pada daerah tertentu diantara frekuensi *cut-off* pertama

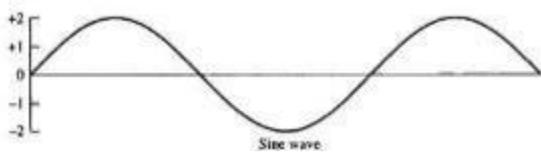
dan frekuensi *cut-off* kedua dan meredam frekuensi diluar daerah tersebut. Dan juga *band pass filter* dikatakan adalah sebuah rangkaian yang dirancang untuk melewatkan frekuensi dalam batasan tertentu dan menolak frekuensi lain diluar frekuensi yang dikehendaki. Sehingga gabungan antara *highpass* dan *lowpass filter* disebut *band pass filter*. *Filter bandpass* akan meneruskan sinyal ke sinyal dengan frekuensi antara (*median frequency*) dan menahan frekuensi dibawah dan diatas median tersebut. Jadi *bandpass filter* adalah piranti yang melakukan tugas memilih sinyal yang diinginkan dari tidak diinginkan berdasarkan frekuensinya. Sinyal yang diinginkan dilewatkan oleh piranti ini sebaik mungkin (tanpa kerugian), sedangkan sinyal yang tidak diinginkan, ditolak untuk melewati piranti filter ini.

## TINJAUAN PUSTAKA

### Sinyal

Sinyal adalah fenomena dari lingkungan yang terukur atau terkuantisasi. (Hanggarsari, Fitriawan, & Yuniati, 2012)

Sinyal analog adalah sinyal pemanfaatan gelombang elektromagnetik. Proses pengiriman suara, misalnya pada teknologi telepon, dilewatkan melalui gelombang elektromagnetik ini. (Hanggarsari et al., 2012).

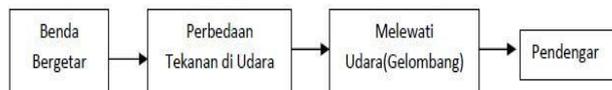


Gambar 1. Contoh Sinyal analog.

Sistem digital merupakan bentuk sampling dari sistem analog. Digital pada dasarnya di code-kan dalam bentuk biner (atau Hexa). Besarnya nilai suatu sistem digital dibatasi oleh lebarnya atau jumlah bit (bandwidth).(Hanggarsari et al., 2012)

### Suara

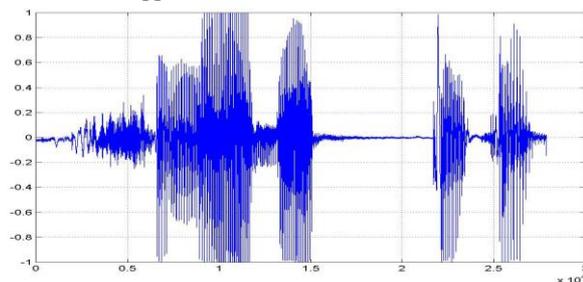
Suara adalah salah satu sinyal yang sangat dipengaruhi frekuensi dan merupakan bentuk sinyal diskrit yang sangat dipengaruhi oleh waktu.(Hanggarsari et al., 2012)



Gambar 2. Proses terjadinya suara (Chris.1996)

### Bentuk Sinyal Wicara

Sinyal wicara merupakan sinyal yang bervariasi lambat sebagai fungsi waktu, dalam hal ini ketika diamati pada durasi yang sangat pendek (5-100ms) karakteristiknya masih stasioner.(Hanggarsari et al., 2012)



Gambar 3. Contoh Sinyal Wicara Ucapan "Selamat Datang"(Hanggarsari et al., 2012)

### Filter (Penyaring)

Filter sering disebut juga dengan rangkaian seleksi frekuensi. Filter berfungsi untuk melewatkan band frekuensi tertentu dan menahannya dari frekuensi diluar band itu. (Azhar, 2018).

Filter mempunyai tiga fungsi dasar yaitu :

1. Pemfilteran (*Filtering*), dengan cara menghilangkan sinyal yang tidak diinginkan.
2. Penghalusan (*smoothing*),
3. Pemrediksi (*prediction*),

Jaringan filter bisa bersifat aktif maupun pasif. Perbedaan dari komponen aktif dan pasif adalah pada komponen aktif dibutuhkan sumber agar dapat bekerja (op-amp dan transistor membutuhkan sumber lagi agar dapat bekerja/digunakan), sedangkan komponen pasif tidak membutuhkan sumber lagi untuk digunakan atau bekerja. (Ii & Pustaka, 1989)

## METODE

### Band Pass Filter

Band pass filter (BPF) adalah rangkaian yang melewatkan frekuensi pada daerah tertentu di antara frekuensi cut-off pertama dan frekuensi cut-off kedua dan meredam frekuensi di luar daerah (2,3 GHz) tersebut.(Jaringan & Ghz, 2016).

*Band Pass Filter* hanya melewatkan sebuah pita frekuensi dan memperlemah semua frekuensi di luar pita itu. Pengertian lain dari *Band Pass Filter* adalah filter yang melewatkan suatu range frekuensi.(Jaringan & Ghz, 2016).



Gambar 4. Diagram blok band pass filter

### FFT (Fast Fourier Transform)

Transformasi *fourier* adalah suatu metode yang sangat efisien untuk menyelesaikan transformasi *fourier* diskrit yang banyak dipakai untuk keperluan analisa sinyal seperti pemfilteran, analisa korelasi, dan analisa spektrum.(Hanggarsari et al., 2012).

*Discrete Fourier Transformasi (DFT)* adalah deretan yang terdefinisi pada kawasan frekuensi – diskrit yang merepresentasikan Transformasi Fourier terhadap suatu deretan terhingga (Finite Duration Sequence).(Hanggarsari et al., 2012)

DFT berperan penting untuk implementasi algoritma suatu varitas pengolahan sinyal, karena efisien untuk komputasi berbagai aplikasi. (Hanggarsari et al., 2012).

Fast fourier transformation atau transformasi fourier cepat, merupakan proses lanjutan dari DFT. Transformasi Fourier ini dilakukan untuk mentransformasikan sinyal dari domain waktu ke domain frekuensi. Hal ini bertujuan agar sinyal dapat diproses dalam spektral substraksi.

## HASIL DAN PEMBAHASAN

Pada source code sebuah program, matlab untuk memfilter sinyal yang bernoise menggunakan metode band pass filter dapat dianalisa bahwa pada program akan menampilkan hasil output berupa 3 figure yaitu grafik sinyal asli, grafik domain waktu, dan grafik domain frekuensi. Sebelumnya adapun beberapa algoritma program filtering sinyal menggunakan filter bandpass yaitu yang pertama menginputkan data sampel dengan format .wav dan selanjutnya data sampel dibaca dengan menggunakan perintah `[f,fs]=audioread('suaraasli(.wav)');`. Dan untuk nama data sampel dapat disesuaikan dengan data sampel yang digunakan. Kemudian perintah `l=length(f);` digunakan hitung panjang dari frekuensi data sampel tersebut. Langkah selanjutnya menampilkan frekuensi dari sinyal asli pada hasil output figure 1. Dan selanjutnya menampilkan hasil konversi dari sinyal asli ke domain waktu, dimana didapat dari frekuensi persekond dibagi dengan jumlah frekuensi. Dan untuk mengubah domain waktu ke domain frekuensi membutuhkan algoritma FFT (fast fourier transform). Dimana fungsi algoritma FFT itu sendiri adalah dalam proses enkripsi, sinyal informasi yang berupa sinyal analog (domain waktu) dikonversikan ke dalam domain frekuensi menggunakan algoritma Fast Fourier Transform (FFT).Sebelum menampilkan figure 3 domain frekuensi perlu diperhatikan sesuaikan atau ganti batas bawah bandpass (beginFreq=7000) dan batas atas bandpass (endFreq =12000). Setelah itu menampilkan domain frekuensi yang didapat dari penggabungan dari jumlah frekuensi dengan nilai absolut algoritma FFT.Lalu program selesai.

Pada data sampel yang saya gunakan adalah data suara gitar. Pada data sampel gitar adalah suara gitar yang telah diubah rate frekuensinya menjadi 8000,16000,32000,48000,96000.

Ternyata pada program matlab bandpass filter tersebut tidak dapat menampilkan nilai matriks pada frekuensi 8000 dan 16000. Dikarenakan metode band pass filter tidak cocok untuk frekuensi yang rendah seperti 8000 dan 16000 melainkan metode lowpass filter cocok untuk menampilkan frekuensi nilai matriks pada sinyal dengan frekuensi 8000 da 16000.

Dan metode band pass filter cocok digunakan untuk sinyal dengan frekuensi diatas 16000 seperti pada percobaan ini frekuensi yang cocok untuk band pass filter untuk 32000,48000dan 96000. Dan nilai matriks yang dihasilkan setiap frekuensi 32000, 48000, dan 96000 berbeda-beda .Ketika dijalankan program filtering sinyal menggunakan band pass filter menghasilkan sebuah nilai matriks pada commad windows yang didapat dari data sampel yang digunakan.

Nilai matriks pada frekuensi 32000 tersebut terdapat :

```

b =
0.1137 0 -0.7956 0 2.3867 0 -3.9778 0 3.9778 0 -
2.3867 0 0.7956 0 -0.1137
a =
1.0000 -2.9532 1.1914 3.0185 -0.3651 -4.1898
0.5777 2.7222 0.0936 -1.2750 -0.1836 0.3306
0.0837 -0.0382 -0.0129

```

Nilai matriks pada frekuensi 48000 tersebut terdapat :

```

b =
0.0119 0 -0.0833 0 0.2500 0 -0.4166 0 0.4166 0 -0.2500
0 -0.0119
a =
1.0000 -6.7558 20.7256 -39.1306 52.3817 -53.9360
44.1867 -28.8584 15.0155 -6.2562 2.0457 -0.4933
0.0851 -0.0106 0.0004

```

Nilai matriks pada frekuensi 96000 tersebut terdapat :

```

b =
0.0002 0 -0.0017 0 0.0050 0 -0.0083 0 0.0083 0 -
0.0050 0 0.0017 0 -0.0002
a =
1.0000 -10.4848 51.2572 -155.0352 324.4379
576.0665
-512.5134 351.9735 -185.6975 74.0892 -21.6754
4.3952 -0.5529 0.0325

```

## HASIL PENGUJIAN

No.	Nama Sampel	Frekuensi Sampel	Keterangan
1.	Suara Gitar1	8000	Length -4000 sampai 4000 x10 <sup>4</sup>
		16000	Length -8000 sampai 8000 x10 <sup>4</sup>
		32000	Length -2 sampai 2 x10 <sup>4</sup>
		48000	Length -2.5 sampai 2.5 x10 <sup>4</sup>
		96000	Length -5 sampai 5 x10 <sup>4</sup>

## KESIMPULAN

1. Pada proses enkripsi untuk mengkonversi sebuah sinyal informasi berupa sinyal analog atau domain waktu kedalam domain frekuensi menggunakan namanya algoritma *Fast Fourier Transform* (FFT).
2. Perintah  $l=length(f)$ ; digunakan hitung panjang dari frekuensi data sampel tersebut.
3. Kegunaan dari program filtering sinyal menggunakan metode band pass filter adalah misalnya sinyal sampel gitar mempunyai sinyal suara dengan frekuensi 0 – 20000 Hz, namun karena suatu hal (misalnya noise) kita cuma butuh sinyal di frekuensi 1000 – 4000 Hz saja, maka bisa menggunakan script matlab / script program filter bandpass.
4. Berdasarkan hasil percobaan diatas bahwa bandpass tidak dapat digunakan untuk menampilkan nilai matriks pada sinyal dengan frekuensi 8000 dan 16000 jadi metode bandpass tidak cocok dengan frekuensi sinyal 8000 dan 16000 tetapi nilai matriks pada frekuensi 8000 dan 16000 akan mengeluarkan nilainya apabila menggunakan metode lowpass dikarenakan lowpass cocok untuk rate frekuensi yang rendah seperti 8000 dan 16000.
5. Pada program matlab bandpass filter tersebut bahwa dapat disimpulkan bahwa nilai matriks yang dihasilkan pada setiap frekuensi berbeda-beda nilai matriks yang didapatkan.
6. Dari hasil percobaan diatas bahwa jika menggunakan frekuensi tinggi maka hasil amplitude dari frekuensi tersebut semakin renggang tetapi sebaliknya jika menggunakan frekuensi rendah maka sinyal akan semakin padat dan mengurangi resiko adanya noise yang terdapat pada sinyal tersebut.
7. Dari hasil output nilai matriks program matlab filtering sinyal menggunakan band pass filter disimpulkan bahwa semakin tinggi frekuensi maka semakin besar nilai matriks pada frekuensinya dan sebaliknya semakin kecil frekuensinya kecil nilai matriks pada frekuensinya.

## REFERENSI

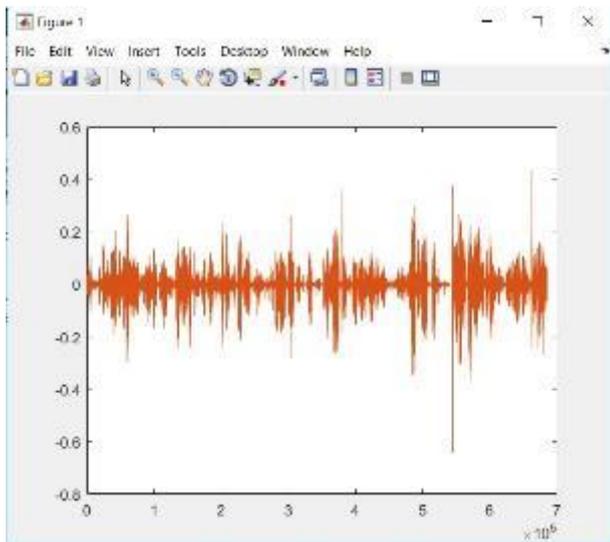
- [1] Azhar, K. (2018). IMPLEMENTASI PERBEDAAN SINGNAL OUTPUT DAN INPUT TERHADAP RESPON FREKUENSI PADA, 7.
- [2] Hanggarsari, P. N., Fitriawan, H., & Yuniati, Y. (2012). SIMULASI SISTEM PENGACAKAN SINYAL SUARA SECARA REALTIME BERBASIS FAST FOURIER TRANSFORM ( FFT ), (3), 192–198.
- [3] Ii, B. A. B., & Pustaka, T. (1989). No Title, 6–22.
- [4] Jaringan, P., & Ghz, W. (2016). DESIGN BAND PASS FILTER UNTUK MENGATASI INTERFERENSI, 4, 52–60.

## LAMPIRAN-LAMPIRAN

### Script pada matlab untuk mengelola sinyal analog dari suara gitar asli

```
clear all;
[a, Fs]=audioread('S4.wav');
plot(a)
handles.a = a;
handles.Fs = Fs;
sound (a, Fs);
```

### Gambar hasil output sinyal analog dari suara gitar asli



### Script pada matlab untuk mengurangi *noise* pada sinyal asli

```
clear all;
[f,fs]=audioread('S1.wav');
l=length(f);
figure(1)
plot(f)

N = size(f,1);
figure(2);
subplot(2,1,1);
stem(1:N, f(:,1));
df=fs/N;
w=(-(N/2):(N/2)-1)*df;
y=fft(f(:,1),N)/N;
y2=fftshift(y);
figure(3)
plot(w,abs(y2));
n=7;
beginFreq = 700/(fs/2);
endFreq = 12000/(fs/2);
[b,a] = butter(n,[beginFreq,endFreq],'bandpass')
%figure(3)
```

### Gambar hasil output sinyal analog telah difilter menggunakan band pass filter

