

Penapisan Sinyal Suara Berderau Menggunakan Tapis Adaptif Finite Impulse Response pada File External Wav

Wisnu Widiarto
Informatika

Fakultas MIPA UNS

Jl.Ir Sutami No.36 A Kertingan
Surakarta

bethoro_wisnu@uns.ac.id

ABSTRAK

Sinyal suara sangatlah penting dalam bidang komunikasi. Namun demikian terdapat sinyal suara yang sudah tercampur dengan gangguan yang berupa noise, sehingga sinyal menjadi lemah atau bahkan menjadi hilang. Agar sinyal suara tersebut dapat diterima secara baik, maka perlu dilakukan penapisan terhadap sinyal yang diterima.

Dalam penelitian ini dilakukan penapisan sinyal suara yang sudah tercampur dengan derau menggunakan tapis adaptif finite impulse response. Aplikasi tersebut diterapkan terhadap file eksternal berekstensi wav, kemudian dilakukan pembangkitan terhadap suara asli, hasil filter dan errornya. Penapisan dilakukan dengan tapis adaptif metode finite impulse response.

Hasil Penapisan diperoleh suatu 'Grafik' maupun 'Suara' yang mendekati Grafik dan Suara Asli. Namun demikian perbandingan Suara ASLI dan HASIL PENAPISAN tetap diperoleh sedikit ERROR dengan Grafik Perbandingan antara suara 'ASLI', suara 'hasil FILTER' dan 'ERROR'. Filter adaptif dapat digunakan untuk menapis berbagai sinyal yang mengandung derau dengan mengatur nilai parameter filter yaitu μ dan N .

Keywords

Noise, tapis adaptif, finite impulse response, file wav.

1. PENDAHULUAN

Sinyal suara seringkali tercampur oleh pengaruh derau. Untuk membantu memperoleh sinyal suara yang bebas dari pengaruh derau, sering digunakan perangkat tambahan yang dinamakan tapis. Tapis biasanya bekerja dengan cara melewatkan sinyal dengan karakteristik berbentuk gelombang, frekuensi, fase dan amplitude tertentu. Tapis dapat diperoleh dengan cara membangkitkan sinyal yang serupa dengan derau dan mengurangkannya dengan sinyal yang bercampur dengan derau. Dengan demikian akan diperoleh sinyal yang terbebas dari pengaruh derau. Salah satu tapis yang dapat digunakan untuk menekan derau pada sinyal informasi adalah tapis adaptif yaitu tapis digital yang menggunakan umpan balik untuk menemukan nilai koefisien tapis terbaik.

Dengan menggunakan algoritma pemrograman Matlab akan disusun suatu simulasi yang dapat menapis suara berderau (berupa file eksternal bertipe wav) sehingga dengan algoritma penapisan filter adaptif model Tapis Adaptif Finite Impulse Response akan dapat dihasilkan suatu suara hasil penapisan yang mendekati suara asli sebelum terkena noise. Pembacaan file eksternal dengan memanfaatkan fungsi wavread pada Matlab kemudian ditapis dengan algoritma program matlab dan hasil penapisan diwujudkan dalam bentuk numeris dan sebuah

file wav yang akan dibandingkan dengan suara asli sebelum terkena noise (derau).

Akan diujicobakan sebuah filter adaptif terhadap sinyal suara yang sudah tercampur oleh adanya derau (noise), dan menguji algoritma hasil penapisan, dengan menggunakan nilai μ yang bervariasi sedangkan nilai N yang tetap, atau dengan nilai N yang bervariasi tetapi nilai μ yang tetap. Nilai μ menunjukkan bobot dari suatu filter dan N menunjukkan panjang dari filter oleh karena itu untuk nilai μ yang relatif lebih kecil dari yang lainnya akan menghasilkan keluaran yang lebih baik karena nilai μ merupakan pengali terhadap error-nya. Sedangkan untuk nilai N yang relatif besar dari yang lainnya akan menghasilkan keluaran yang lebih baik karena terkait dengan lebar jendela pemfilteran. Sedangkan untuk nilai L yang sangat besar dalam proses filtering memerlukan banyak memori karena nilai L berhubungan dengan cacah data yang difilter, sehingga dalam percobaan ini digunakan hanya $L=1000$, tetapi untuk selengkapnya dengan L sepanjang ukuran ordonya ($L=3473$) dapat dilihat dalam lampiran.

Dengan memanfaatkan sound editing, rekaman suara dapat terbentuk berupa file dengan type wav. Hasil file eksternal tersebut dibaca dengan fungsi pemrograman matlab, kemudian dilakukan penapisan atas derau yang dikenakan pada suara aslinya. Transfer nilai numeris yang tersimpan dalam sebuah variabel dimensi matrik ke dalam bentuk file bertipe wav dapat ditangani oleh function yang disediakan oleh pemrograman Matlab, sehingga tidak terlalu disulitkan oleh pembuatan fungsi terlebih dahulu.

2. LANDASAN TEORI

2.1 Matrix Laboratory

Bahasa pemrograman MATLAB (kepanjangan MATrix LABoratory) merupakan software paket matematika yang bertumpu pada matriks, yang menyediakan fasilitas-fasilitas operasi matriks dan vektor secara sistematis dan mudah dalam penggunaannya [1]. MATLAB menyediakan fasilitas pemakaian user secara interaktif maupun secara pemrograman. Secara interaktif, karena Matlab menyediakan prompt yang dapat digunakan sebagai pelaksanaan eksekusi perintah. Secara pemrograman, karena Matlab juga dilengkapi dengan fasilitas yang dapat dipakai untuk menyusun program. Matlab juga sudah dilengkapi dengan berbagai function yang dapat langsung dipanggil (secara interaktif maupun secara pemrograman), sehingga fasilitas tersebut dapat mempermudah pelaksanaan penyusunan program. Fungsi-fungsi yang berkaitan misalnya fungsi wavread yang digunakan untuk pemanggilan file eksternal bertipe wav, fungsi wavwrite yang digunakan untuk penulisan file dengan type wav, demikian juga fungsi size yang digunakan untuk mendeteksi panjang orde sebuah suara yang direkam lewat sound editing dengan file bertipe wav. Fungsi-

fungsi built-in milik Matlab (yang telah disediakan) sebagai besar juga merupakan file yang disimpan dengan type .m, fungsi-fungsi tersebut yang nantinya akan digunakan dalam penelitian ini adalah wavread, wavwrite, size, plot dan subplot.

2.2 Filter

Filter atau tapis merupakan suatu elemen yang sangat penting dalam sinyal prosesing. Filter memiliki berbagai klasifikasi, salah satunya dapat digolongkan berdasarkan jenis frekuensi yang dapat dilewatkan (frekuensi yang dapat ditapis) oleh filter tersebut. Jenis ini dikenal LPF (Low Pass Filter) merupakan filter yang dapat melewatkan komponen frekuensi rendah dari suatu sinyal, BPF (Band Pass Filter) merupakan suatu filter yang dapat melewatkan komponen frekuensi tengah, dan HPF (High Pass Filter) yang merupakan filter yang hanya dapat melewatkan komponen frekuensi tinggi. Disamping masih banyak jenis filter yang lainnya seperti filter Butherwort atau filter Kalman atau yang lainnya [2].

Memfilter sinyal sinus yang mengandung tiga buah frekuensi yaitu 5, 15 dan 30 hz, dalam hal ini frekuensi sinyal yang akan difilter adalah frekuensi 5 dan 30 Hz. Tujuannya adalah menguji algoritma apakah hasil penapisannya benar-benar 15 Hz. Sinyal sinus dibentuk menggunakan fungsi sinus berikut:

1. S1 = sin (2*pi*5*t) untuk f =5 Hz
2. S2 = sin (2*pi*15*t) untuk f =15 Hz
3. S3 = sin (2*pi*30*t) untuk f =30 Hz ; 0 < t < 1 detik.

Dengan menjumlahkan ketiga buah sinyal tersebut diperoleh sinyal sinus yang mengandung tiga komponen frekuensi. Pembangkitan panjang Orde dilakukan dengan function yang sudah tersedia dalam pemrograman MatLab dan hasilnya ditampung dalam sebuah variabel, misalnya variabel L.

```
orde = size(wavread('w_mlab.wav'));
L = orde(1);
```

Pemanggilan file external yang berupa file suara berekstensi .wav yang tersimpan dalam file w_mlab.wav dilakukan dengan function wavread dan ditampung dalam variabel v yang berupa digit angka sebagai berikut :

```
v=wavread('w_mlab',L);
```

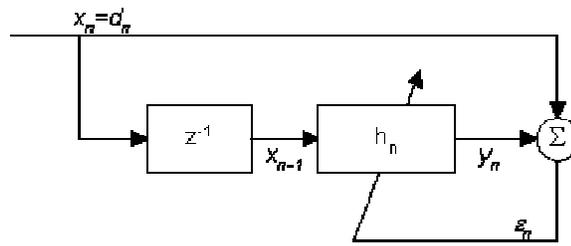
Dengan mempergunakan persamaan filter FIR, persamaan algoritma adaptif prediktif yang terbentuk sebagai berikut (L = panjang orde; n = data ke):

$$y(n) = h(n,0) * x(n-1) + h(n,1) * x(n-2) + \dots + h(n,L-1) * x(n-L)$$

$$e(n) = x(n) - y(n)$$

$$h(n+1, N-1) = h(n, N-1) + 2 * \mu * x(n-N-2) * e(n)$$

Filter yang akan digunakan dapat berbentuk bermacam-macam diantaranya adalah filter adaptif yang memiliki model sebagaimana gambar berikut : (Xn adalah Sinyal masukan)



Gambar 1. Filter Adaktif

2.3 Sound Editing

Beberapa tahun yang lalu, para pemakai PC sudah cukup puas dengan PC Speaker yang hanya dapat mengeluarkan nada tunggal tertentu pada suatu saat. Kemudian berubah muncul sebuah Interface Card Audio (sound card) dengan kemampuan 8 bit (nomor dari bit ini menunjukkan banyaknya sample point) dimana semakin besar atau banyak nilai sample point tersebut, maka akan semakin baik pula kualitas suara yang dapat dihasilkannya. Sound card mempunyai peranan yang penting, yaitu pemanfaatan segala indera manusia terutama mata dan telinga. secara umum alat tambahan yang ditempatkan pada komputer ini mempunyai fungsi untuk mengolah data suara dalam bentuk analog dan merubahnya kedalam bentuk digital dan disimpan ke dalam sebuah file bertipe data suara.

Waveform audio adalah sebuah standard data suara yang dihasilkan dari bentuk analog kedalam bentuk digital. Porses dan cara pembuatan tipe data ini adalah dengan memasukkan informasi sebuah signal suara yang melalui mic input untuk proses perekaman suara yang berasal dari lingkungan luar, atau melalui line input jika merekam suara yang sumbernya dari line out, seperti CD audio, tape, suara dari video. Kedua jalur masukan ini biasanya sudah disediakan dalam sound card. Kemudian sound card menghitung identitasnya dalam saat - saat yang telah ditentukan. Untuk kualitas dari suara ditentukan oleh sample rate atau banyaknya sampel yang diambil dalam 1 detik.

Dalam dunia kedokteran fonokardiogram digunakan untuk menganalisa kondisi jantung pasien. Fono kardiogram berupa visualisasi suara jantung yang berupa isyarat berosilasi sangat cepat (frekuensi tinggi). Pada kasus kelainan bising jantung terdapat pola-pola fonokardiogram (frekuensi rendah). Penelitian ini dilakukan usaha untuk mendeteksi inti isyarat suara jantung yang merupakan sampul dari isyarat yang berosilasi sangat cepat dengan menggunakan transformasi hilbert, tapis lolos bawah dan tapis rerata. Hasil penelitian menunjukkan bahwa transformasi hilbert bisa diaplikasikan untuk deteksi sampul, namun perlu dilakukan penelitian lebih lanjut untuk menentukan tapis-tapis yang optimal [3].

Implementasi tapis digital secara waktu nyata (real-time) tidaklah mudah karena dibatasi oleh beberapa kendala semisal kebutuhan akan komputasi yang cepat, kesalahan kuantisasi pada ADC-DAC, kesalahan kuantisasi koefisien filter dan kesalahan pada pembulatan aritmatika [4]. Implementasi sebuah filter digital bandpass dengan metode penjendelaan Blackman ke dalam perangkat keras berupa Digital Signal Processor (DSP) TMS320C6711, menunjukan bahwa filter tersebut dapat bekerja secara real-time dan memenuhi spesifikasi yang telah ditentukan.

Noise dapat menyebabkan bunyi atau sinyal suara menjadi terganggu, menjadi lemah atau hilang, sehingga diperlukan proses penguatan suara terhadap noise. Untuk mempermudah

proses penguatan data suara terhadap tingkat noise, digunakan sistem track record suara pada proses perekaman dengan menggunakan perangkat lunak Cakewalk dan SonicFoundry. Hasilnya dianalisis untuk memperoleh level dan jenis noise, kemudian data diperbaiki menggunakan metode Flat-Top Sampling [5].

3. PEMBAHASAN

3.1 Pembangkitan Suara Asli

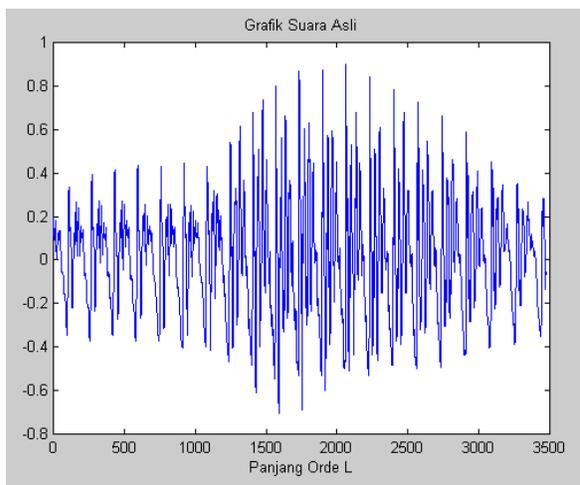
Dalam penelitian ini dilakukan dengan cara merekam data berupa suara yang dihasilkan dari sound editing dengan kata suara 'LAB'. Sound perekaman tersebut diedit sedemikian rupa sehingga suara yang dihasilkan diedit sesuai yang diinginkan dan tidak ada lagi jenjang antara awal sebelum suara muncul dan jenjang antara akhir setelah suara. Pembangkitan panjang Orde dilakukan dengan function yang sudah tersedia dalam pemrograman MatLab dan hasilnya ditampung dalam sebuah variabel, misalnya variabel L.

```
orde = size(wavread('w_mlab.wav'));
```

```
L = orde(1);
```

Pemanggilan file external yang berupa file suara berekstensi .wav yang tersimpan dalam file w_mlab.wav dilakukan dengan function wavread dan ditampung dalam variabel v yang berupa digit angka sebagai berikut :

```
v=wavread('w_mlab',L);
```



Gambar 2. Grafik Suara Asli yang dihasilkan dari Sound Recording (belum terkena noise/derau)

Program yang tersusun untuk pemanggilan file suara w_mlab.wav yang diwujudkan dalam bentuk grafik adalah sebagai berikut :

```
orde = size(wavread('w_mlab.wav'));
L = orde(1);
k=1:L;
v=wavread('w_mlab',L);
##### download file w_mlab.wav ###
plot(k,v);
title('Grafik Suara Asli');
xlabel('Panjang Orde L');
```

3.2 Pembangkitan Suara Noise

Dalam penelitian ini, derau (noise) dibentuk dengan simulasi yang dibangkitkan dan diciptakan sendiri. Derau (noise) tersebut akan dicampurkan dengan pembangkitan suara asli, yang akan menghasilkan simulasi suara yang terkena derau (noise). Pembangkitan panjang Orde dilakukan dengan function yang sudah tersedia dalam pemrograman MatLab dan hasilnya ditampung dalam variabel yang diinginkan, misal variabel L.

```
orde = size(wavread('w_noise.wav'));
```

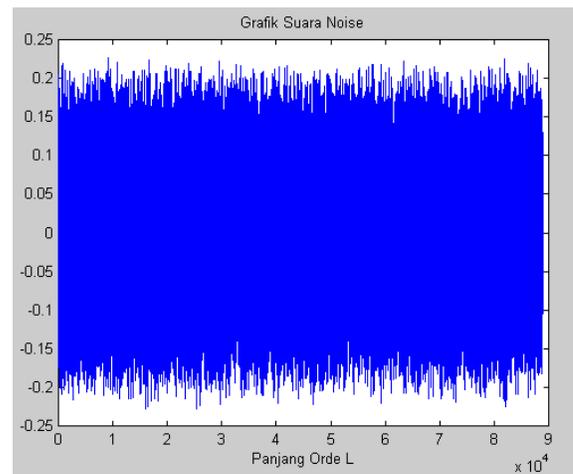
```
L = orde(1);
```

Pemanggilan file external yang berupa file suara berekstensi .wav yang tersimpan dalam file w_noise.wav dilakukan dengan function wavread dan ditampung dalam variabel r yang berupa digit angka sebagai berikut :

```
r=wavread('w_noise',L);
```

Program yang tersusun untuk pemanggilan file suara w_noise.wav yang diwujudkan dalam bentuk grafik adalah seperti dalam program di bawah, dengan orde sepanjang L.

```
orde = size(wavread('w_noise.wav'));
L = orde(1);
k=1:L;
r=wavread('w_noise',L);
##### download file w_noise.wav ###
plot(k,r);
title('Grafik Suara Noise');
xlabel('Panjang Orde L');
```



Gambar 3. Grafik Suara yang dibuat sebagai campuran mix (sebagai noise/derau)

3.3 Pembangkitan Suara Mix (Suara Terkena Derau)

Derau (noise) yang sudah dibentuk dengan simulasi yang dibangkitkan dan diciptakan sendiri dicampurkan dengan suara asli yang dibangkitkan, menghasilkan simulasi suara yang terkena derau (noise). Pembangkitan panjang Orde dari suara miks yang sudah terkena derau dilakukan dengan function yang sudah tersedia dalam pemrograman MatLab dan hasilnya ditampung dalam sebuah variabel L.

```
orde = size(wavread('w_mlabmix.wav'));
```

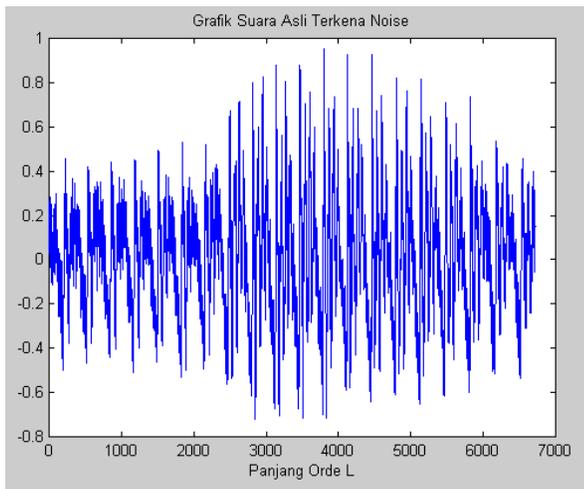
```
L = orde(1);
```

Pemanggilan file external yang berupa file suara berekstensi .wav yang tersimpan dalam file w_mlabmix.wav dilakukan dengan function wavread dan ditampung dalam variabel x yang berupa digit angka sebagai berikut :

```
x=wavread('w_mlabmix',L);
```

Program selengkapnya yang tersusun untuk pemanggilan file suara w_mlabmix.wav yang diwujudkan dalam bentuk grafik dapat dilihat dalam struktur berikut :

```
orde = size(wavread('w_mlabmix.wav'));
L = orde(1);
k=1:L;
x=wavread('w_mlabmix',L);
##### download file w_mlabmix.wav #####
plot(k,x);
title('Grafik Suara Asli Terkena Noise');
xlabel('Panjang Orde L');
```



Gambar 4. Grafik Suara Rekaman yang sudah terkena noise (di-mix dengan rekaman noise/derau)

3.4 Pembangkitan Hasil Filter (Dari Suara Terkena Noise)

Dari suara yang terkena derau (noise) yang dibentuk lewat simulasi, maka dilakukan penapisan, dengan tujuan untuk memperoleh suara hasil penapisan yang sesuai dengan suara asli yang telah dibangkitkan melalui perekaman sound recording. Dengan mempergunakan persamaan filter FIR, persamaan algoritma adaptif prediktif yang terbentuk sebagai berikut (L = panjang orde; n = data ke):

$$y(n) = h(n,0) * x(n-1) + h(n,1) * x(n-2) + \dots + h(n,L-1) * x(n-L)$$

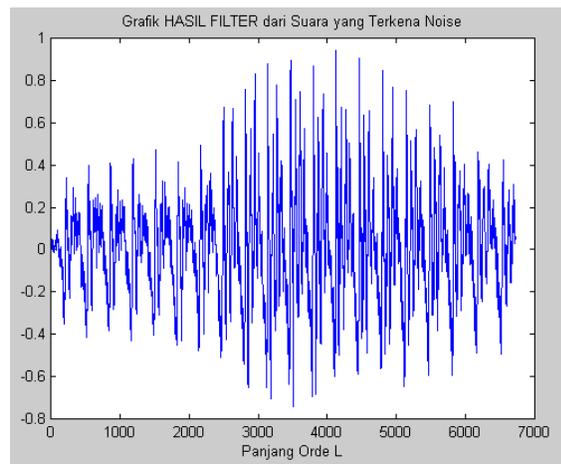
$$e(n) = x(n) - y(n)$$

$$h(n,1:N,1) = h(n,1:N,1) + 2 * \mu * x(n-N-2) * e(n)$$

```
orde = size(wavread('w_mlabmix.wav'));
L = orde(1);
mu = 0.1;
N = 3;
k = 1:L;
```

```
[x,FS,NBITS]=WAVREAD('w_mlabmix',L)
% file dibaca adalah 'w_mlabmix.wav'
% sample rate (FS) in Hertz
% the number of bits per sample (NBITS) used to
encode the data in the file.

d = x;
for i=1:N
    h(i,(N+1))=0;
end
for n=(N+1):L
    z(n)=0;
    for i=1:N
        y(n)=h(i,n)*x(n-i)+z(n);
        z(n)=y(n);
    end
    e(n)=d(n)-y(n);
    for i=1:N
        h(i,n+1)=h(i,n)+2*mu*e(n)*x(n-i);
    end
end
plot(k,y);
title('Grafik HASIL FILTER dari Suara yang Terkena
Noise');
xlabel('Panjang Orde L');
WAVWRITE(y,FS,NBITS,'wHs_filt.wav')
% file terfilter yang terbentuk adalah 'wHs_filt.wav'
```



Gambar 5. Grafik Suara Hasil Filtering / Penapisan (penapisan atas rekaman yang terkena noise/derau)

3.5 Pembangkitan Error Hasil Filter (Dari Suara Terkena Noise)

Dalam melakukan penapisan, setelah itu apapun, biasanya masih terdapat error kesalahan, besarnya kesalahan adalah:

```
e(n)=d(n)-y(n);
[v,FS,NBITS]=WAVREAD('w_mlab',L)
[x,FS,NBITS]=WAVREAD('w_mlabmix',L)
d = v;
for i=1:N
    h(i,(N+1))=0;
end
for n=(N+1):L
```

```

z(n)=0;
for i=1:N
    y(n)=h(i,n)*x(n-i)+z(n);
    z(n)=y(n);
end
e(n)=d(n)-y(n);
for i=1:N
    h(i,n+1)=h(i,n)+2*mu*e(n)*x(n-i);
end
end

```

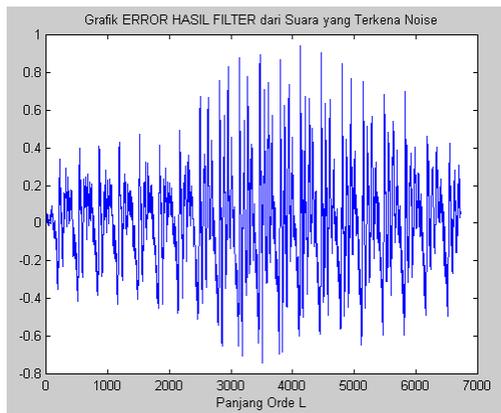
Sehingga program selengkapnya adalah sebagai berikut:

```

orde = size(wavread('w_mlabmix.wav'));
L = orde(1);
mu = 0.1;
N = 3;
k = 1:L;
[x,FS,NBITS]=WAVREAD('w_mlabmix',L)

d = x;
for i=1:N
    h(i,(N+1))=0;
end
for n=(N+1):L
    z(n)=0;
    for i=1:N
        y(n)=h(i,n)*x(n-i)+z(n);
        z(n)=y(n);
    end
    e(n)=d(n)-y(n);
    for i=1:N
        h(i,n+1)=h(i,n)+2*mu*e(n)*x(n-i);
    end
end
plot(k,y);
title('Grafik ERROR HASIL FILTER dari Suara yang
Terkena Noise');
xlabel('Panjang Orde L');
WAVWRITE(e,FS,NBITS,'wHs_err.wav')
% file error yang terbentuk adalah 'wHs_err.wav'

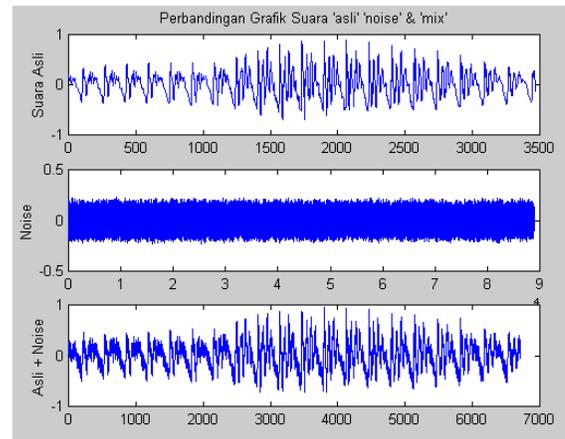
```



Gambar 6. Grafik Error Suara Hasil Filtering / Penapisan (hasil penapisan dibandingkan dengan rekaman asli)

3.6 Pembangkitan Perbandingan Suara Asli-Noise-Mix (Dari Suara Terkena Noise)

Hasil pembangkitan perbandingan suara asli – noise – mix terlihat dalam grafik pada gambar 7.



Gambar 7. Grafik Perbandingan Suara Asli – Noise – Suara Mix (perbandingan dengan panjang orde L)

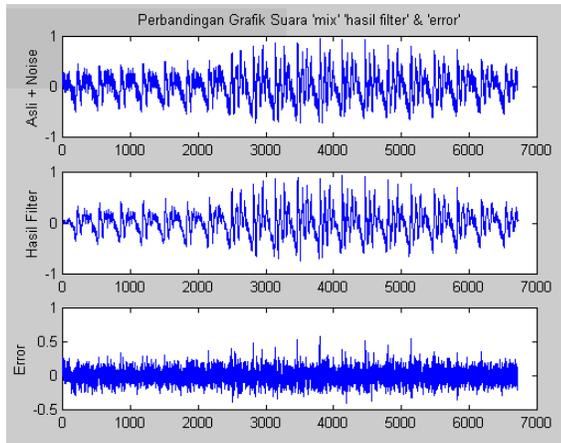
```

% untuk Grafik suara ASLI
orde = size(wavread('w_mlab.wav'));
L = orde(1);
i = 1:L;
v = wavread('w_mlab',L);
% untuk Grafik suara NOISE
orde = size(wavread('w_noise.wav'));
L = orde(1);
j = 1:L;
r = wavread('w_noise',L);
% untuk Grafik suara MIX
orde = size(wavread('w_mlabmix.wav'));
L = orde(1);
k = 1:L;
x = wavread('w_mlabmix',L);
% untuk menggambarkan GRAFIK
subplot(3,1,1); plot(i,v);
title('Perbandingan Grafik Suara "asli" "noise" &
"mix"');
ylabel('Suara Asli');
subplot(3,1,2); plot(j,r); ylabel('Noise');
subplot(3,1,3); plot(k,x); ylabel('Asli + Noise');

```

3.7 Pembangkitan Perbandingan Suara Mix Terkena Noise – Hasil Filter - Error

Hasil pembangkitan perbandingan suara mix terkena noise – hasil filter – Error terlihat dalam grafik pada gambar 8.



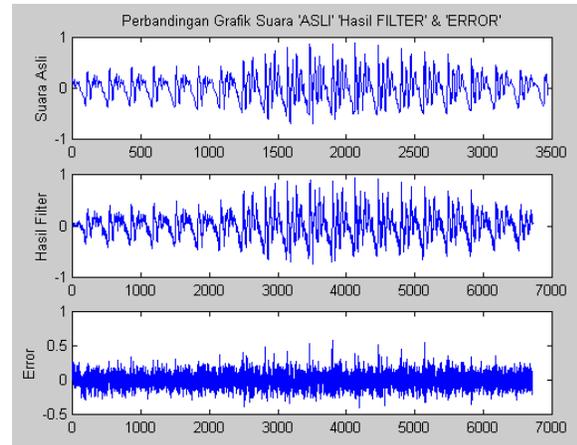
Gambar 8. Grafik Perbandingan Suara Mix – Hasil Filter – Error (perbandingan dengan panjang orde L)

Sedangkan programnya adalah sebagai berikut:

```
% untuk Grafik suara MIX
orde = size(wavread('w_mlabmix.wav'));
L = orde(1);j = 1:L;
x = wavread('w_mlabmix',L);
% untuk Grafik suara HASIL FILTER & ERROR
orde = size(wavread('w_mlabmix.wav'));
L = orde(1);
mu = 0.1;
N = 3;k = 1:L;
[x,FS,NBITS]=WAVREAD('w_mlabmix',L)
% file dibaca adalah 'w_mlabmix.wav'
% sample rate (FS) in Hertz
% the number of bits per sample (NBITS) used to
encode the data in the file.
d = x;
for i=1:N
    h(i,(N+1))=0;
end
for n=(N+1):L
    z(n)=0;
    for i=1:N
        y(n)=h(i,n)*x(n-i)+z(n);
        z(n)=y(n);
    end
    e(n)=d(n)-y(n);
    for i=1:N
        h(i,n+1)=h(i,n)+2*mu*e(n)*x(n-i);
    end
end
% untuk menggambarkan GRAFIK
subplot(3,1,1); plot(j,x);
title('Perbandingan Grafik Suara "mix" "hasil filter"
& "error"');
ylabel('Asli + Noise');
subplot(3,1,2); plot(k,y); ylabel('Hasil Filter');
subplot(3,1,3); plot(k,e); ylabel('Error');
```

3.8 Pembangkitan Perbandingan Suara Asli (Belum Terkena Noise) – Hasil Filter - Error

Hasil pembangkitan perbandingan suara asli sebelum terkena noise – hasil filter – Error terlihat dalam grafik pada gambar 9.



Gambar 9. Grafik Perbandingan Suara Asli – Hasil Filter – Mix (perbandingan dengan panjang orde L)

Sedangkan program untuk membandingkan Suara Asli – Hasil Filter – Mix (perbandingan dengan panjang orde L) sebagai berikut:

```
% untuk Grafik suara ASLI
orde = size(wavread('w_mlab.wav'));
L = orde(1);
j = 1:L;
v = wavread('w_mlab',L);
% untuk Grafik suara HASIL FILTER & ERROR
orde = size(wavread('w_mlabmix.wav'));
L = orde(1);
mu = 0.1;
N = 3;
k = 1:L;
[x,FS,NBITS]=WAVREAD('w_mlabmix',L)
% file dibaca adalah 'w_mlabmix.wav'
% sample rate (FS) in Hertz
% the number of bits per sample (NBITS) used to
encode the data in the file.
d = x;
for i=1:N
    h(i,(N+1))=0;
end
for n=(N+1):L
    z(n)=0;
    for i=1:N
        y(n)=h(i,n)*x(n-i)+z(n);
        z(n)=y(n);
    end
    e(n)=d(n)-y(n);
    for i=1:N
        h(i,n+1)=h(i,n)+2*mu*e(n)*x(n-i);
    end
end
% untuk menggambarkan GRAFIK
subplot(3,1,1); plot(j,v);
```

```
title('Perbandingan Grafik Suara "ASLI" "Hasil
FILTER" & "ERROR"');
ylabel('Suara Asli');
subplot(3,1,2); plot(k,y); ylabel('Hasil Filter');
subplot(3,1,3); plot(k,e); ylabel('Error');
```

3.9 Hasil Perbandingan Bentuk Numeris

Hasil pembangkitan perbandingan dalam bentuk numeris terlihat dalam table 1

Table 1. Hasil pembangkitan perbandingan dalam bentuk numeris

no	asli	mix	filter	Error	Filt+err
1	0.1953	0.1917	0.0000	0.0000	0.0000
2	0.1641	0.1062	0.0000	0.0000	0.0000
3	0.1406	0.1699	0.0000	0.0000	0.0000
4	0.1172	0.0282	0.0000	0.0000	0.0000
5	0.1016	0.2654	0.0000	0.0000	0.0000
6	0.0938	0.1555	0.0000	0.0000	0.0000
7	0.1016	0.1055	0.0000	0.0000	0.0000
8	0.1094	0.0616	0.0000	0.0000	0.0000
9	0.1328	0.2126	0.0000	0.0000	0.0000
10	0.1406	0.0900	0.0000	0.0000	0.0000
11	0.1406	0.2177	0.0000	0.1406	0.1406
12	0.1406	0.0779	0.0008	0.1399	0.1406
13	0.1563	0.1574	0.0016	0.1546	0.1563
14	0.1719	0.0443	0.0024	0.1694	0.1719
15	0.1797	0.0880	0.0036	0.1760	0.1797
16	0.1563	0.2167	0.0041	0.1522	0.1563
17	0.1172	0.1136	0.0051	0.1121	0.1172
18	0.0781	0.2699	0.0057	0.0724	0.0781
19	0.0625	0.2845	0.0071	0.0554	0.0625
20	0.0469	0.2052	0.0078	0.0391	0.0469
...
...
981	0.0000	-0.0334	-0.0282	0.0282	0.0000
982	0.0313	0.0021	-0.0205	0.0518	0.0313
983	0.0625	-0.0865	-0.0155	0.0780	0.0625
984	0.1016	-0.0270	-0.0118	0.1134	0.1016
985	0.1406	-0.0547	-0.0075	0.1481	0.1406
986	0.1875	0.0189	-0.0009	0.1884	0.1875
987	0.2188	0.0133	0.0007	0.2181	0.2188
988	0.2266	0.0354	0.0046	0.2220	0.2266
989	0.2109	0.1193	0.0116	0.1994	0.2109
990	0.1719	0.0790	0.0104	0.1615	0.1719
991	0.1328	0.1867	0.0087	0.1241	0.1328
992	0.1094	-0.0113	0.0075	0.1018	0.1094
993	0.1094	0.2533	0.0033	0.1061	0.1094
994	0.1172	0.0973	-0.0016	0.1188	0.1172
995	0.1172	0.2919	-0.0082	0.1254	0.1172
996	0.0938	0.2495	-0.0136	0.1074	0.0938
997	0.0781	0.1270	-0.0182	0.0964	0.0781
998	0.0859	0.1041	-0.0268	0.1128	0.0859
999	0.1250	0.1506	-0.0341	0.1591	0.1250
1000	0.1563	0.2407	-0.0353	0.1916	0.1563

4. PENUTUP

Beberapa file eksternal dapat dicoba dibaca dan diolah dengan algoritma Matlab. Dalam penelitian ini yang dimaksud sebagai file eksternal adalah file suara berekstensi wav, yang rekaman suaranya dapat dibuat dengan menggunakan sound recording. Sedangkan untuk pembacaan file eksternal digunakan fungsi wavread :

```
v = wavread('w_mlab,L'); atau
```

```
[x,FS,NBITS]=WAVREAD('w_mlab',L)
```

Dimana L merupakan panjang orde yang dapat diketahui dengan fungsi size.

```
orde = size(wavread('w_mlab.wav'));
```

Hasil penapisan yang berbentuk numeris juga dapat ditransfer dalam bentuk file suara bertipe wav dengan cara menuliskan hasil penapisan yang diujudkan kembali dalam file bertipe wav menggunakan fungsi :

```
wavwrite(x,FS,NBITS,'wHs_filt.wav');
```

Hasil Penapisan diperoleh suatu 'Grafik' maupun 'Suara' yang mendekati Grafik dan Suara Asli. Namun demikian tidak ada sesuatu yang dikatakan sempurna, sehingga perbandingan Suara ASLI dan HASIL PENAPISAN tetap diperoleh sedikit ERROR dengan Grafik Perbandingan antara suara 'ASLI', suara 'hasil FILTER' dan 'ERROR'. Filter adaptif dapat digunakan untuk menapis berbagai sinyal yang mengandung derau dengan mengatur nilai parameter filter yaitu mu dan N.

Penelitian ini hanya berujud simulasi yang mengambil suara asli dengan sound editing berisikan sebuah kata suara 'LAB', dengan noise yang dibentuk sendiri dengan menambahkannya sebagai mixer, namun demikian belum diwujudkan dalam praktek lapangan, sehingga sinyal yang terbentukpun sebagaimana terlihat dalam gambar grafiknya, masih pendek panjang ordenya yaitu 3473.

Sebagai dasar untuk memperluas dan mengembangkan permasalahan, alangkah baiknya algoritma pemrograman dan software tersebut diterapkan sebagai pemantau dan alat penapis terhadap suatu ruangan tertentu, akan lebih sempurna lagi jika diciptakan suatu alat yang dapat mengkontinukan hardware dan software tersebut sebagai alat khusus penapis derau (noise). Pemrograman Matlab hanya sebagai salah satu sarana untuk mengelola algoritma program, maka tidak menutup kemungkinan jika dikembangkan dengan software perangkat lunak lain sebagai pembanding.

5. DAFTAR PUSTAKA

- [1] Penny, J.E.T.; 1995; Numerical Methods Using MATLAB; Ellis Horwood;
- [2] Johnson, D.E.; 1997; Electric Circuit Analysis, third edition; Prentice Hall, Inc; New Jersey
- [3] Widodo,S.B.; 2011; Aplikasi Transformasi Hilbert untuk Deteksi Sampul (Envelope Detection) Isyarat Suara Jantung; Prosiding Konferensi Nasional ICT-M Politeknik Telkom (KNIP)
- [4] Ariyanto, G.; Nurgiyatna; Sudarmilah E.; 2002; Implementasi Filter Digital Finite Impulse Response Metode Penjendelaan Blackman pada DSP TMS320C6711; Jurnal Teknik Elektro Emitor Vol. 2, No. 2, September 2002
- [5] Puspasari, R.; 2010; Analisa dan Perancangan Aplikasi Bantu Perbaikan Signal to Noise Ratio (SNR) dengan Metode Flat-Top Sampling; Jurnal ISSN 1978 - 8282 Vol 3 No. 2 Januari 2010