

MODUL 4

PEMFLITERAN PADA SINYAL WICARA

I. TUJUAN

- Mahasiswa mampu menyusun filter digital dan melakukan pemfilteran pada sinyal wicara

II. DASAR TEORI

2.1. Filter IIR

Yang perlu diingat disini bahwa *infinite impulse response* (IIR) dalam hal ini bukan berarti filter yang bekerja dari nilai negatif tak hingga sampai positif tak hingga. Pengertian sederhana untuk *infinite impulse respon filter* disini adalah bahwa output filter merupakan fungsi dari kondisi input sekarang, input sebelumnya dan output di waktu sebelumnya. Konsep ini kemudian lebih kita kenal sebagai *recursive filter*, yang mana melibatkan proses *feedback* dan *feed forward*. Dalam bentuk persamaan beda yang menghubungkan input dengan output dinyatakan seperti persmaaan (1) berikut ini.

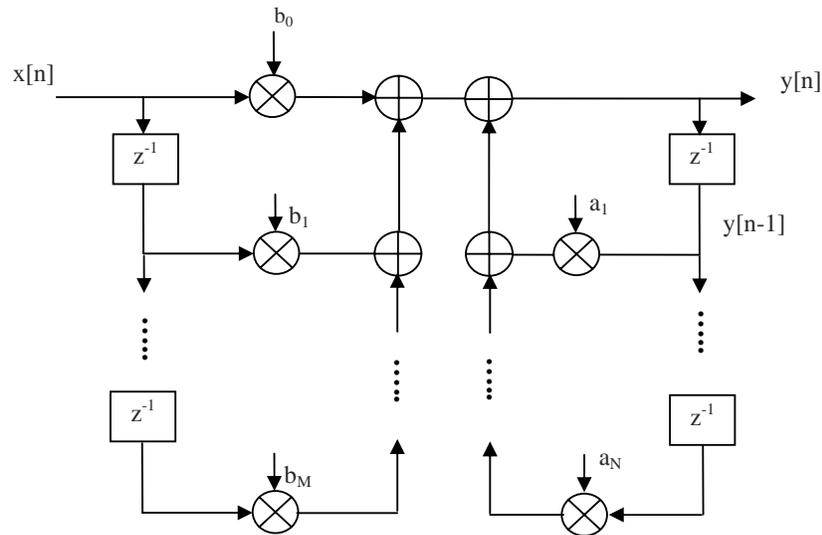
$$y[n] = \sum_{l=1}^N a_l y[n-l] + \sum_{k=0}^M b_k x[n-k] \quad (1)$$

dimana:

- $\{b_k\}$ koefisien *feed forward*
- $\{a_l\}$ koefisien *feed back*
- banyaknya (total koefisien) = M+N+1
- N ditetapkan sebagai orde filter IIR

Untuk merealisasikan ke dalam sebuah program simulasi atau perangkat keras maka bentuk persamaan diatas dapat disederhanakan ke dalam diagram blok Gambar 1.

Untuk implementasi sebuah *low pass filter* bersifat *narrow-band* menggunakan sebuah filter IIR merupakan pilihan yang sangat sulit tetapi masih mungkin dilakukan. Satu alasannya adalah penentuan orde yang tepat sehingga menghasilkan bentuk yang tajam pada respon frekuensi relative sulit. Pada domain *unit circle* bidang-z sering ditandai dengan letak *pole-pole* yang ada diluar lingkaran, hal ini secara fisis memberikan arti bahwa filter yang dihasilkan tidak stabil.



Gambar 1. Diagram blok Filter IIR

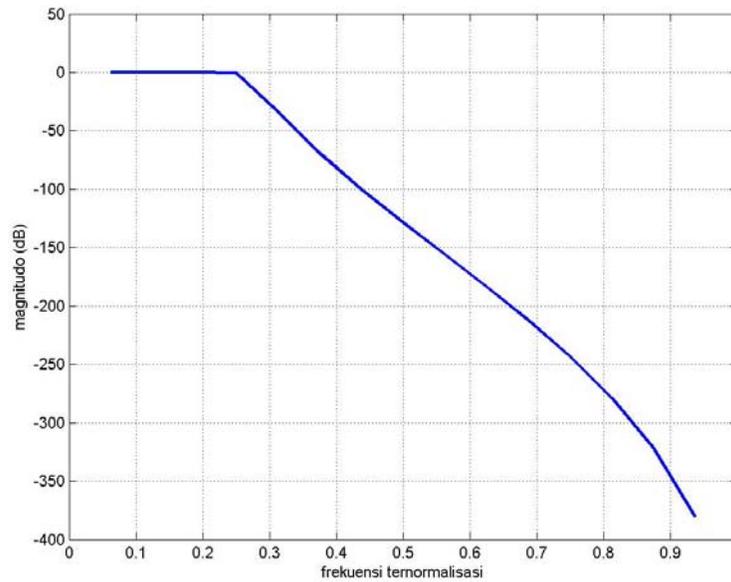
Kita coba untuk merealisasikan dalam program Matlab secara sederhana dengan melihat pada masing-masing kasus, dalam hal ini adalah *low pass filter* (LPF) dan *high pass filter* (HPF).

Contoh 1:

Kita akan mencoba merancang sebuah *low pass filter* (LPF) IIR dengan memanfaatkan filter *Butterworth*. Frekuensi cut off ditetapkan sebesar 2000 Hz. Dalam hal ini frekuensi sampling adalah 10000 Hz. Langkah realisasi dalam Matlab adalah sebagai berikut.

```
clear all;
R=0.2;
N=16;
Wn=0.2;
figure(1);
[B,A] = butter(N,Wn);
[H,w]=freqz(B,A,N);
len_f=length(H);
f=1/len_f:1/len_f:1;
plot(f,20*log10(abs(H)), 'linewidth', 2)
```

Dari langkah ini akan didapatkan respon frkuensi seperti gambar berikut.



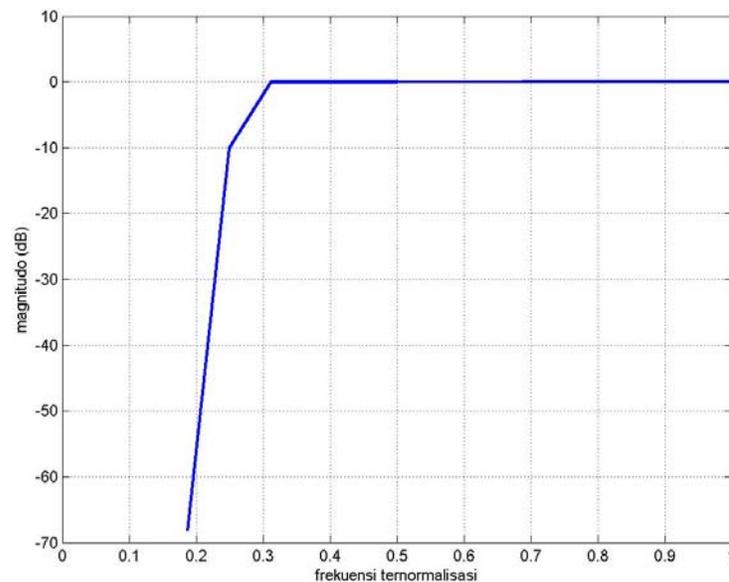
Gambar 2. Respon Frekuensi Filter IIR, LPF

Contoh 2:

Pada contoh kedua ini kita akan mencoba merancang sebuah filter IIR untuk *high pass filter* (HPF). Tetap dengan frekuensi *cut off* 2000 Hz, dan frekuensi sampling 10000. Langkah pemrogramannya adalah dengan sedikit memodifikasi bagian berikut.

```
[B,A] = butter(N,Wn,'high');
```

Ini akan memberikan respon frekuensi seperti berikut.



Gambar 3. Respon Frekuensi Filter IIR, HPF

2.2. Filter FIR

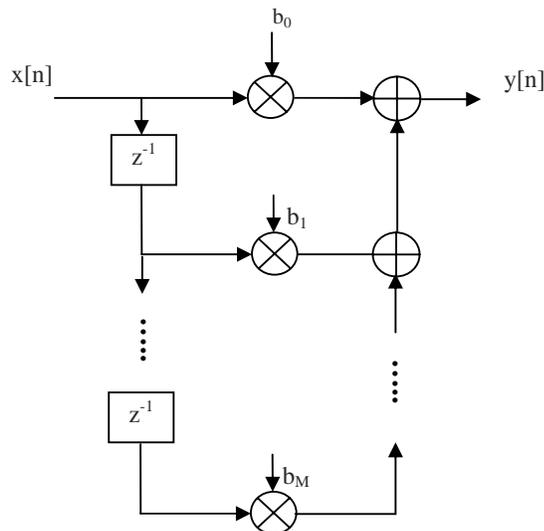
Sebuah *finite impulse respon filter* (filter FIR) memiliki hubungan input dan output dalam domain waktu diskrit sebagai berikut:

$$y[n] = \sum_{k=0}^M b_k x[n-k] = \sum_{k=0}^M h[k] x[n-k] \quad (2)$$

dimana:

- $\{b_k\}$ = koefisien *feed forward*
- banyaknya (total koefisien) $L = M + 1$
- M ditetapkan sebagai orde filter FIR

Dalam realisasi diagram blok akan dapat digambarkan seperti pada Gambar 4 berikut ini



Gambar 4. Diagram blok FIR Filter

Untuk tujuan simulasi perangkat lunak kita bisa memanfaatkan fungsi standar berikut ini: $B = \text{FIR1}(N, W_n)$

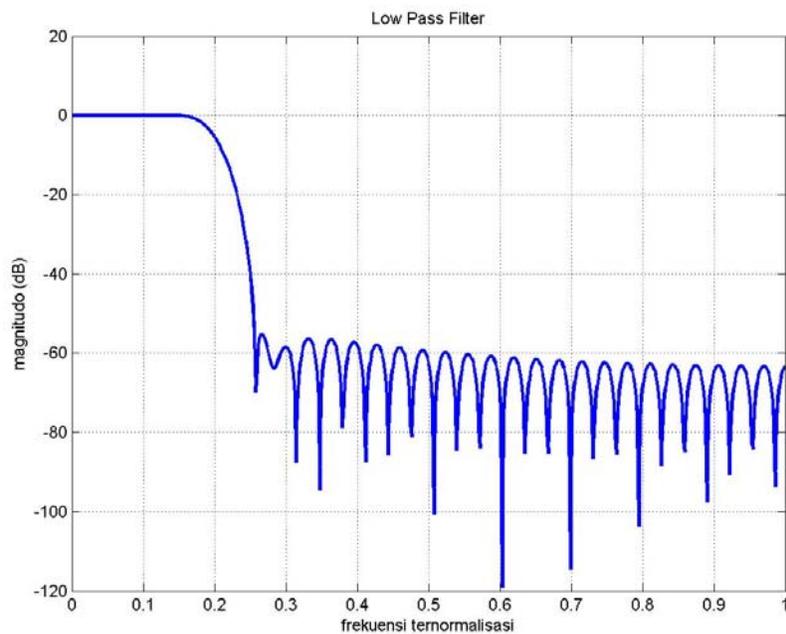
Ini merupakan sebuah langkah untuk merancang filter digital FIR dengan orde sebesar N , dan frekuensi cut off W_n . Secara *default* oleh Matlab ditetapkan bahwa perintah tersebut akan menghasilkan sebuah *low pass filter* (LPF). Perintah ini akan menghasilkan koefisien-koefisien filter sepanjang $(N+1)$ dan akan disimpan pada vektor B . Karena dalam domain digital, maka nilai frekuensi *cut off* harus berada dalam rentang $0 < W_n < 1.0$. Nilai 1.0 akan memiliki ekuivalensi dengan nilai 0,5 dari *sampling rate* ($fs/2$). Yang perlu anda ketahui juga adalah bahwa B merupakan nilai real dan memiliki fase yang linear. Sedangkan *gain* ternormalisasi filter pada W_n sebesar -6 dB.

Contoh 3:

Kita akan merancang sebuah LPF dengan frekuensi *cut off* sebesar 2000 Hz. Frekuensi sampling yang ditetapkan adalah 10000 Hz. Orde filter ditetapkan sebesar 32. Maka langkah pembuatan programnya adalah sebagai berikut:

```
fs=10000;  
[x,fs]=wavread('a.wav');  
Wn = .20;  
N = 32;  
LP = fir1(N,Wn);  
[H_x,w]=freqz(LP);  
len_f=length(H_x);  
f=1/len_f:1/len_f:1;  
plot(f,20*log10(abs(H_x)))  
grid
```

Hasilnya adalah respon frekuensi seperti Gambar 5 berikut



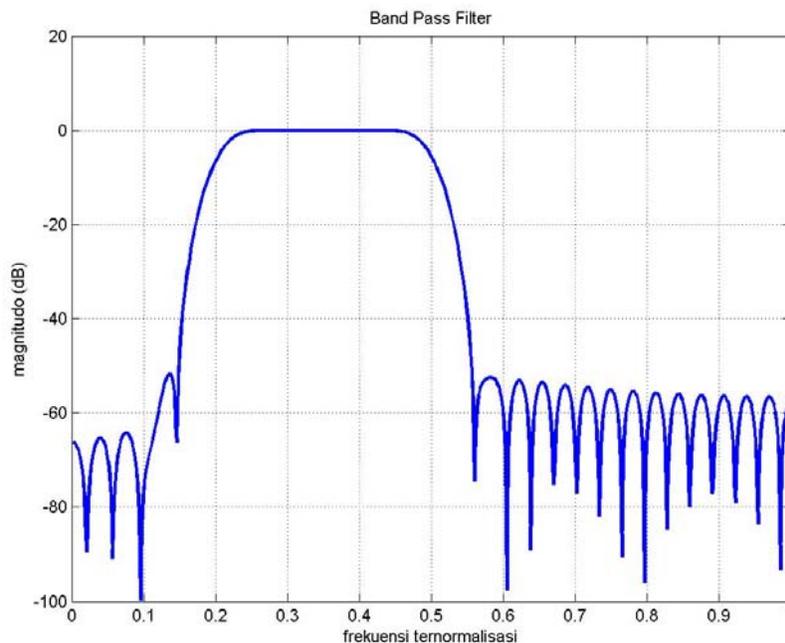
Gambar 5. Respon Frekuensi Low Pass Filter

Contoh 4:

Kita akan merancang sebuah *Band Pass Filter* (BPF) dengan frekuensi *cut off* sebesar 2000 Hz (untuk daerah rendah) dan 5000 Hz (untuk daerah tinggi). Frekuensi sampling yang ditetapkan adalah 10000 Hz. Orde filter ditetapkan sebesar 32. Beberapa bagian program diatas perlu modifikasi seperti berikut.

```
Wn1 = [.20, .50];  
BP = fir1(N,Wn1);
```

Hasilnya akan didapatkan respon frekuensi seperti pada Gambar 6 berikut ini.



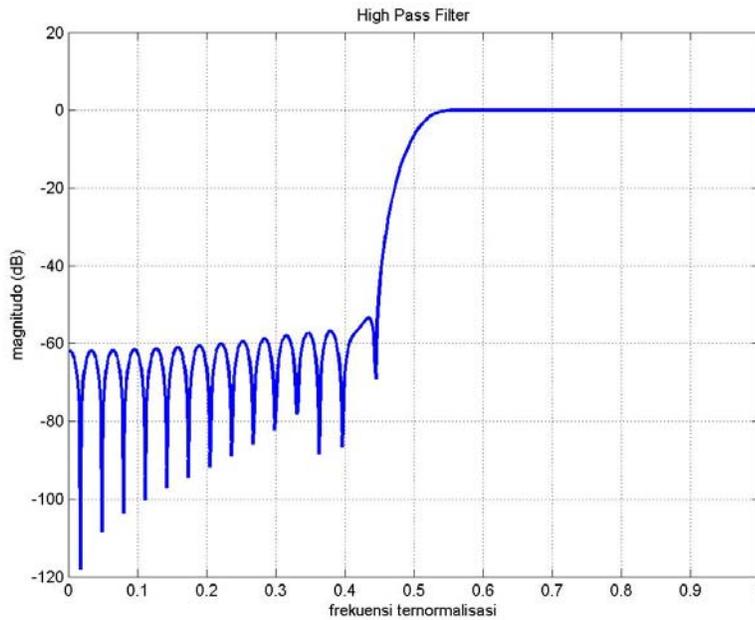
Gambar 6. Respon Frekuensi Band Pass Filter

Contoh 5:

Kita akan merancang sebuah *High Pass Filter* (HPF) dengan frekuensi *cut off* sebesar 5000 Hz (untuk daerah tinggi). Frekuensi sampling yang ditetapkan adalah 10000 Hz. Orde filter ditetapkan sebesar 32. Beberapa bagian program diatas perlu modifikasi seperti berikut.

```
Wn2 = .50;  
HP = fir1(N,Wn2,'high');
```

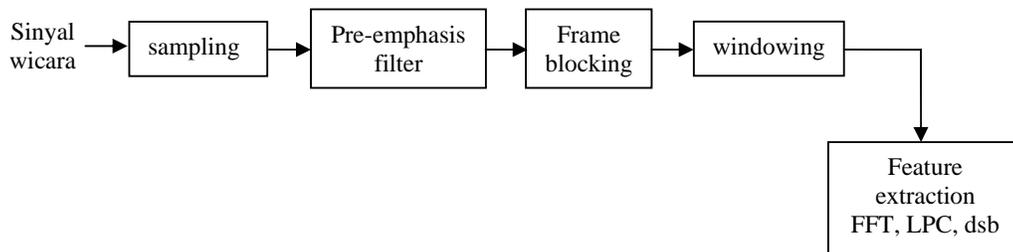
Hasilnya berupa akan didapatkan respon frekuensi seperti pada Gambar 7 berikut ini.



Gambar 7 Respon Frekuensi High Pass Filter

2.3. Filter Pre-Emphasis

Dalam proses pengolahan sinyal wicara *pre emphasis* filter diperlukan setelah proses sampling. Tujuan dari pemfilteran ini adalah untuk mendapatkan bentuk *spectral* frekuensi sinyal wicara yang lebih halus. Dimana bentuk *spectral* yang relatif bernilai tinggi untuk daerah rendah dan cenderung turun secara tajam untuk daerah fekuensi diatas 2000 Hz.



Gambar 8. Posisi Filter *Pre-Emphasis* pada sistem pengolah wicara

Filter *pre-emphasis* didasari oleh hubungan input/output dalam domain waktu yang dinyatakan dalam persamaan beda seperti berikut:

$$y(n) = x(n) - ax(n-1) \quad (3)$$

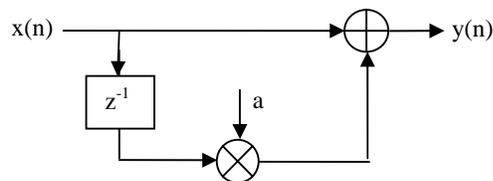
dimana:

a merupakan konstanta filter *pre-emphasis*, biasanya bernilai $0.9 \leq a \leq 1.0$

Dalam bentuk dasar operator z sebagai unit filter, persamaan diatas akan memberikan sebuah *transfer function* filter *pre-emphasis* seperti berikut.

$$H(z) = 1 - az^{-1} \quad (4)$$

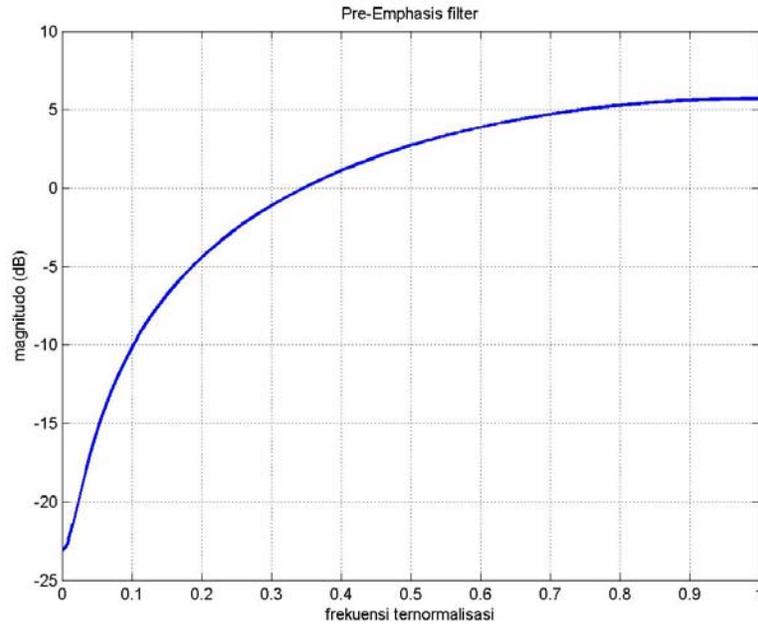
Bentuk ini kemudian akan memberikan dasar pembentukan diagram blok yang menggambarkan hubungan input dan output seperti pada Gambar 8.



Gambar 9. Diagram blok *pre-emphasis* filter

Dengan memanfaatkan perangkat lunak Matlab kita akan dengan mudah mendapatkan bentuk respon frekuensi filter pre-empasis.

```
clear all;
w=0:.01:3.14;
a=0.93;
H=1-a*exp(-j*w);
plot(w/3.14,20*log10(abs(H)), 'linewidth', 2)
grid
axis([0 1.00 -25 10])
xlabel('frekuensi ternormalisasi')
ylabel('magnitudo (dB)')
title('Pre-Emphasis filter')
```



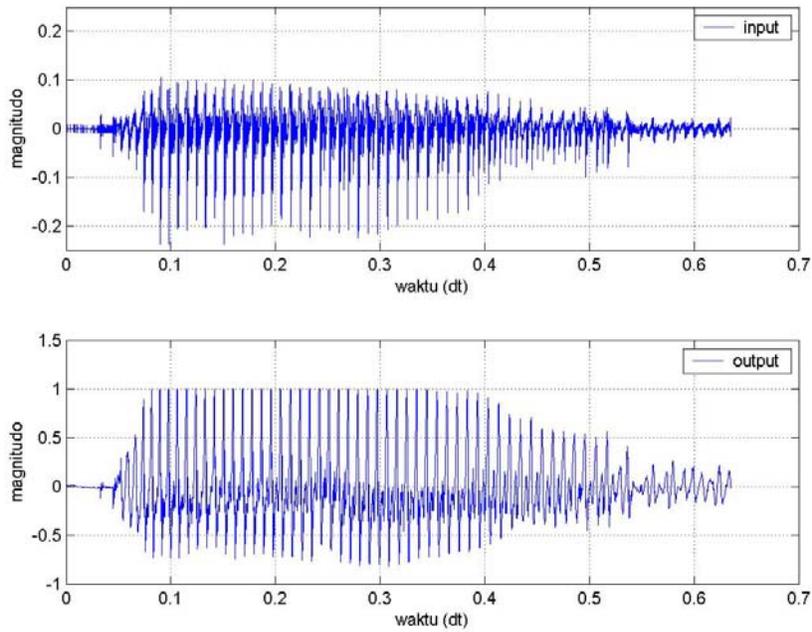
Gambar 10. Respon frekuensi filter pre-emphasis

Dengan nilai $a = 0,93$ akan mampu melakukan penghalusan *spectral* sinyal wicara yang secara umum mengalami penurunan sebesar 6 dB/octav.

Sekarang yang menjadi pertanyaan adalah bagaimana pengaruh sebenarnya filter ini pada sebuah sinyal wicara? Untuk itu anda dapat memanfaatkan program dibawah ini.

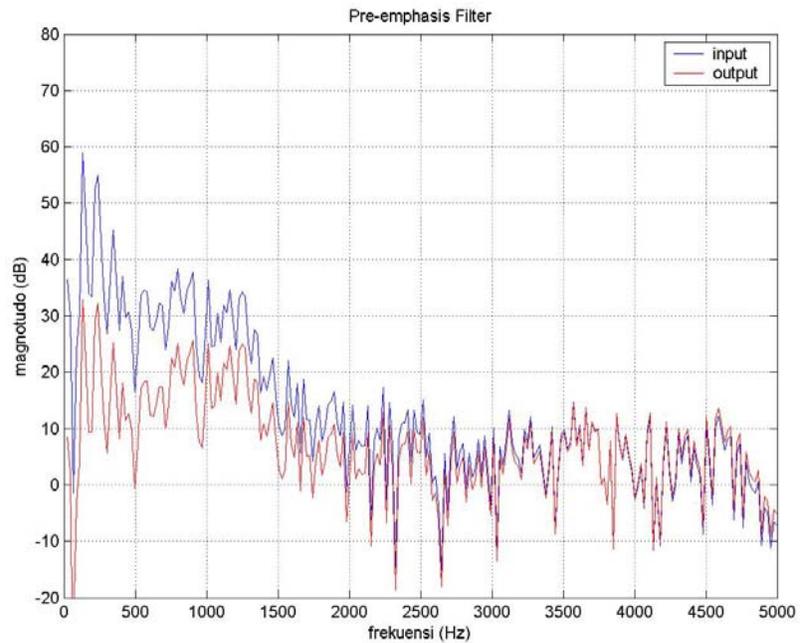
```
clear all;
fs=10000;[x,fs]=wavread('a.wav');
xx=length(x)+1;x(xx)=0; alpha=0.96;
for i=2:xx
    y0(i)=x(i-1);
end
for i=1:xx
    y(i) = x(i) - alpha*y0(i);
end
subplot(211)
t=1:xx;
plot(t/fs,y);legend('input');grid
xlabel('waktu (dt)'); ylabel('magnitudo');axis([0 0.7 -0.25 0.25]);
subplot(212)
plot(t/fs,y0); legend('output');grid
xlabel('waktu (dt)'); ylabel('magnitudo'); axis([0 0.7 -1 1.5])
```

Hasilnya adalah berupa sebuah gambaran bentuk sinyal input dan output dari file sinyal wicara 'a.wav' dalam domain waktu.



Gambar 11. Sinyal input dan output dari *pre-emphasis* filter dalam domain waktu

Sedangkan hasil yang didapatkan dalam bentuk domain frekuensi adalah seperti berikut.



Gambar 12. Sinyal input dan output dari *pre-emphasis* filter dalam domain frekuensi

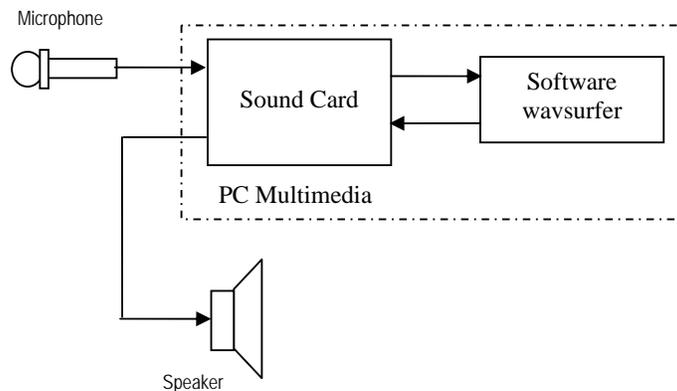
III. PERANGKAT YANG DIPERLUKAN

- 1 (satu) buah *PC Multimedia* lengkap *sound card* dan *microphone*
- Satu perangkat lunak Matlab

IV. LANGKAH-LANGKAH PERCOBAAN

4.1. Penataan Perangkat

Sebelum melakukan percobaan harus dilakukan penataan seperti pada Gambar 13 berikut ini.



Gambar 13. Penataan perangkat percobaan *filtering* sinyal wicara

PC harus dilengkapi dengan peralatan multimedia seperti *sound card*, speaker aktif dan *microphone*. Untuk *microphone* dan speaker aktif bisa juga digantikan dengan *head set* lengkap. Sebelum anda memulai praktikum, sebaiknya dites dulu, apakah seluruh perangkat *multimedia* anda sudah terintegrasi dengan PC.

4.2. Pemfilteran Sinyal Wicara dengan IIR

Pada bagian ini akan dilakukan perancangan filter IIR, mengamati bentuknya respon frekuensi, melakukan pemfilteran pada sinyal wicara, dan melihat pengaruhnya dalam domain waktu dan domain frekuensi. Langkah-langkahnya adalah seperti berikut:

1. Rancang sebuah *low pass filter* IIR dengan spesifikasi seperti berikut

$f_c = 4000$ Hz, frekuensi sampling $f_s = 10000$ Hz, dan pembentukan filter didasarkan pada metode yang sederhana yaitu *Butterworth filter*.

2. Amati bentuk respon frekuensi yang dihasilkannya, usahakan untuk menampilkan dengan sumbu mendatar frekuensi (Hz) dan sumbu tegak berupa magnitudo dalam besaran dB.
3. Tambahkan program untuk memanggil sebuah file '*.wav', dalam hal ini bisa digunakan hasil perekaman yang telah dilakukan pada percobaan 1, misal file 'u.wav'.

```
[s,fs]=wavread('FILE_U.wav');
```
4. Lakukan pemfilteran dengan menggunakan koefisien-koefisien IIR yang telah dirancang pada langkah 1.

```
s0=filter(B,A,s);
```
5. Buat sebuah tambahan program untuk mengamati bentuk spectralnya, bandingkan sinyal wicara tersebut sebelum pemfilteran dan sesudah pemfilteran.
6. Coba tambahkan *noise Gaussian* dengan *varians* $\sigma^2 = 0.2$ pada sinyal wicara yang dipanggil pada langkah ke 3. Dan lakukan proses pemfilteran ulang seperti pada langkah 4.
7. Ulangi langkah 5 untuk melihat pengaruh *noise* pada spectral sinyal wicara, dan lihat pengaruh pemfilteran pada spectral sinyal bernois tersebut.
8. Setelah menyelesaikan langkah 1 sampai 7, cobalah untuk membuat sebuah filter *high pass filter* IIR dengan spesifikasi yang sama dengan yang telah dilakukan dengan *low pass filter*, kecuali frekuensi *cut off* dirubah menjadi $f_c = 200$ Hz.

4.3. Pemfilteran Sinyal Wicara dengan FIR

Pada bagian ini akan dilakukan perancangan filter FIR, mengamati bentuknya respon frekuensi, melakukan pemfilteran pada sinyal wicara, dan melihat pengaruhnya dalam domain waktu dan domain frekuensi. Langkah-langkahnya adalah seperti berikut:

1. Rancang sebuah *low pass filter* FIR dengan spesifikasi seperti berikut:
 $f_c = 4000$ Hz, frekuensi sampling $f_s = 10000$ Hz, dan pembentukan filter didasarkan pada metode yang sederhana yaitu seperti pada contoh yang ada di bagian teori.
2. Amati bentuk respon frekuensi yang dihasilkannya, usahakan menampilkan dengan sumbu mendatar frekuensi (Hz) dan sumbu tegak berupa magnitudo dalam besaran dB.
3. Tambahkan program untuk memanggil sebuah file '*.wav', dalam hal ini bisa kita gunakan hasil perekaman yang telah dilakukan pada percobaan 1, misal file 'FILE_U.wav'.

```
[s,fs]=wavread('FILE_U.wav');
```

4. Lakukan pemfilteran dengan menggunakan koefisien-koefisien FIR yang telah dirancang pada langkah 1

$$y_1 = \text{conv}(LP, x);$$
5. Buat sebuah tambahan program untuk mengamati bentuk spectralnya, bandingkan sinyal wicara tersebut sebelum pemfilteran dan sesudah pemfilteran.
6. Coba tambahkan noise Gaussian dengan varians $\sigma^2 = 0.2$ pada sinyal wicara yang dipanggil pada langkah ke 3. Dan lakukan proses pemfilteran ulang seperti pada langkah 4.
7. Ulangi langkah 5 untuk melihat pengaruh noise pada spectral sinyal wicara, dan lihat pengaruh pemfilteran pada spectral sinyal tersebut.
8. Setelah menyelesaikan langkah 1 sampai 7, cobalah untuk membuat sebuah filter *high pass filter* FIR dengan spesifikasi yang sama dengan yang telah dilakukan dengan *low pass filter*, kecuali *frekuensi cut off* anda rubah menjadi $f_c = 200$ Hz.
9. Ulangi langkah 8 dengan band pass filter yang memiliki spesifikasi $f_L = 200$ Hz dan $f_H = 4000$ Hz

4.4. Pre Emphasis Filter pada Sinyal Speech

Pada bagian ini akan dilakukan perancangan filter *pre-emphasis*, mengamati bentuknya respon frekuensi, melakukan pemfilteran pada sinyal wicara, dan melihat pengaruhnya dalam domain waktu dan domain frekuensi. Langkah-langkahnya adalah seperti berikut:

1. Lakukan perancangan filter *pre-emphasis* dengan nilai $a = 0,95$. Dalam hal ini kita bisa memanfaatkan program yang ada pada bagian teori.
2. Panggil sebuah file '*.wav', dalam hal ini bisa memilih file 'aiueo.wav' atau jika tidak keberatan lakukan perekaman lebih dulu dan anda gunakan sebagai file '*.wav' yang akan dianalisa.
3. Amati bentuk *spectral* sinyal, sebelum dan sesudah diperlakukan dengan *pre-emphasis*.
4. Cobalah rancang sebuah filter *de-emphasis*. Jika mengalami kesulitan dengan hal ini, tanyakan kepada instruktur atau dosen praktikum.

5. ANALISA DATA DAN TUGAS

1. Anda telah melakukan serangkaian percobaan dengan filter digital untuk sinyal wicara, yang harus dilakukan sekarang adalah membuat analisa tentang filter digital tersebut dan jawab berbagai masalah berikut:
 - a. Mengapa pemfilteran sinyal wicara dilakukan pada daerah 200 Hz sampai 4000 Hz?
 - b. Apa pengaruh filter pre-emphasis pada bentuk spectral sinyal wicara?
 - c. Seberapa besar redaman yang dihasilkan sebuah filter FIR dan filter IIR yang telah anda rancang terhadap spectral sinyal wicara tepat pada daerah frekuensi *cut-off*?
2. Berikan analisa tentang sinyal wicara bernoise, dan pengaruh filter low pass yang dirancang.