

**TKE 2403**

# **SISTEM PENGOLAHAN ISYARAT**

**Kuliah 10 – Filter Digital**



**UNIVERSITAS  
MERCU BUANA  
YOGYAKARTA**

**Indah Susilawati, S.T., M.Eng.**

**Program Studi Teknik Elektro  
Fakultas Teknik dan Ilmu Komputer  
Universitas Mercu Buana Yogyakarta**

**2009**

## KULIAH 10

# SISTEM PENGOLAHAN ISYARAT FILTER DIGITAL

### Alihragam Z

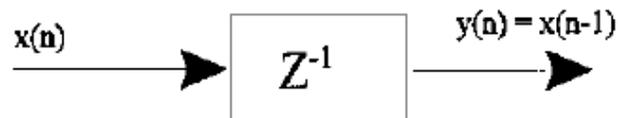
Persamaan untuk alihragam Z mirip dengan persamaan alihragam Fourier, yaitu:

$$X(z) \triangleq Z[x(n)] = \sum_{n=-\infty}^{\infty} x(n) Z^{-n} \quad (1)$$

Dengan  $z$  adalah sebarang variabel kompleks. Jika diidentifikasi dengan sebuah barisan data (seperti  $x(n)$ ),  $z^{-n}$  menyatakan pergeseran sebanyak  $n$  sampel, atau sama dengan pergeseran waktu sepanjang  $nT_s$  detik. Sifat pergeseran dari  $z^{-n}$  dapat dinyatakan sebagai,

$$Z[x(n - k)] = z^{-k} Z(x(n)) \quad (2)$$

Sebagai contoh, karakteristik alihragam Z dapat digunakan untuk proses tunda satu satuan seperti diperlihatkan pada Gambar 1. Pada proses yang digambarkan tsb maka output sistem adalah sama dengan inputnya namun tertunda satu sampel.



Gambar 1. Proses tunda satu satuan yang menggeser data sebanyak satu data (sampel). Pangkat  $n$  yang lain dapat digunakan untuk tundaan yang lebih besar.

### Fungsi Transfer Filter Digital

Fungsi tranfer didefinisikan sebagai,

$$H(z) = \frac{Y(z)}{X(z)} \quad (3)$$

Maka untuk proses yang digambarkan pada Gambar 1, fungsi transfernya adalah  $H(z) = z^{-1}$ . Namun sebagian besar fungsi transfer mempunyai bentuk yang lebih kompleks, melibatkan polinomial dalam  $z$  pada pembilang dan penyebutnya, Perhatikan persamaan berikut.

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Nz^{-N}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_Dz^{-D}} \quad (4)$$

Orde penyebut yaitu  $D$  dapat kurang, sama, atau lebih besar daripada orde pembilang yaitu  $N$ . Sistem dengan orde penyebut sama dengan 1 akan lebih stabil daripada orde yang lebih tinggi.

Dari definisi  $H(z)$  dapat ditentukan output untuk input yang telah tertentu, dan dapat dinyatakan sebagai,

$$Y(z) = H(z) X(z) = X(z) \frac{\sum_{k=0}^{N-1} b(k) z^{-k}}{\sum_{l=0}^{D-1} a(l) z^{-l}} \quad (5)$$

Dengan menerapkan interpretasi pergeseran waktu pada persamaan (5) maka persamaan input – output dapat dinyatakan sbb:

$$y(n) = \sum_{k=0}^K b(k) x(n - k) - \sum_{l=0}^L a(l) y(n - l) \quad (6)$$

Jika diinginkan spektrum frekuensi  $H(z)$ , dapat dihasilkan dari persamaan (5) dengan sedikit modifikasi yaitu substitusi  $z = e^{j\omega}$ ,

$$H(m) = \frac{Y(m)}{X(m)} = \frac{\sum_{n=0}^{N-1} b(n) e^{(-j2\pi mn/N)}}{\sum_{n=0}^{D-1} a(n) e^{(-j2\pi mn/N)}} = \frac{\text{fft}(b_n)}{\text{fft}(a_n)} \quad (7)$$

Dengan fft adalah alihragam Fourier. Sebagaimana dalam alihragam Fourier, frekuensi dapat diperoleh dari variabel m dengan cara mengalikannya dengan  $f_s/N$  atau  $1/(NT_s)$ .

$$\begin{aligned}\omega &= 2\pi f \\ &= 2\pi m/N\end{aligned}$$

### Contoh 1

Temukan dan gambarkan karakteristik frekuensi dan tanggapan impuls dari proses yang mempunyai fungsi transfer sbb,

$$H(z) = \frac{0.2 + 0.5z^{-1}}{1 - 0.2z^{-1} + 0.8z^{-2}}$$

### Penyelesaian

Matlab banyak menyediakan fungsi untuk perancangan dan aplikasi filter yang juga dapat digunakan untuk analisis fungsi tranfer filter digital. Fungsi “filter” mengimplementasikan filter digital dan juga dapat digunakan untuk implementasi proses yang dinyatakan dalam fungsi tranfer alihragam Z. Fungsi “filter” menentukan output  $y(n)$  terhadap  $x(n)$  untuk proses tersebut sebagaimana dinyatakan dalam koefisien a dan b. Sintaknya dalam Matlab adalah sbb:

$$y = \text{filter}(b, a, x)$$

dengan x adalah input, y adalah output, a dan b adalah koefisien fungsi tranfer yang ditentukan dalam  $H(z)$ . Berikut adalah m *file* untuk contoh 1.

```
% Contoh 1 Bab Filter Digital
% Plot karakteristik frekuensi dan tanggapan impuls
% Asumsi: frekuensi pencuplikan 1 kHz

close all; clear all; clc;
fs = 1000; % Frekuensi pencuplikan
N = 512; % Jumlah titik

% Mendefinisikan koefisien a dan b berdasarkan H(z)
a = [1 -.2 .8]; % Penyebut H(z)
b = [.2 .5]; % Pembilang H(z)

% Plot karakteristik frekuensi H(z) menggunakan fft
H = fft(b,N)./fft(a,N); % Menghitung H(f)
Hm = 20*log10(abs(H)); % Menghitung magnitude H dalam dB
```

```

Theta = (angle(H))*2*pi;      % Menghitung fase H dalam derajat
f = (1:N/2)*fs/N ;           % Vektor frekuensi untuk plotting

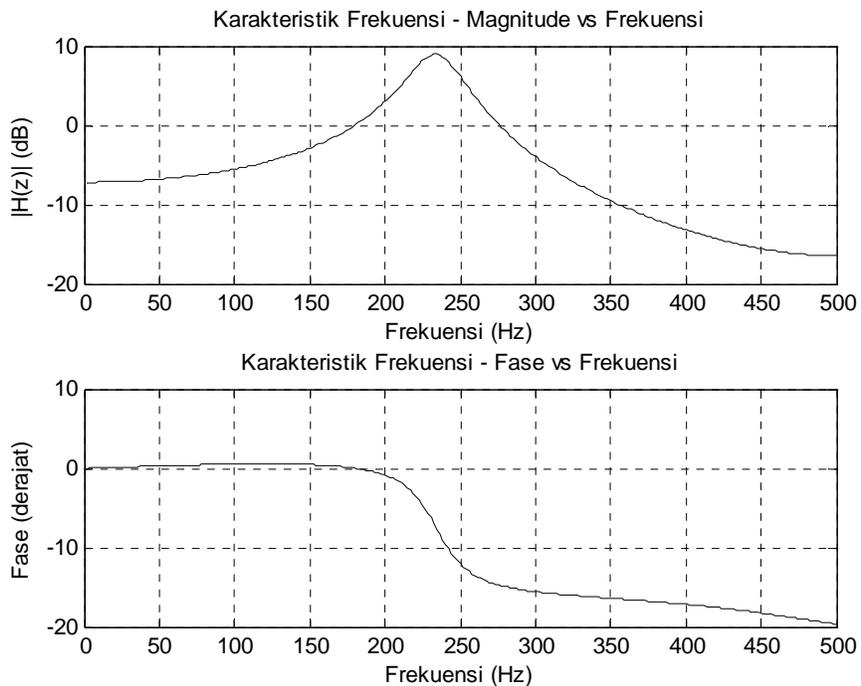
subplot(2,1,1);
plot (f,Hm(1:N/2), 'k');      % Plot magnitudo H
xlabel('Frekuensi (Hz)');
ylabel('|H(z)| (dB)');
grid on;

subplot(2,1,2);
plot(f,Theta(1:N/2), 'k');    % Plot fase H
xlabel('Frekuensi (Hz)');
ylabel('Fase (derajat)');
grid on;

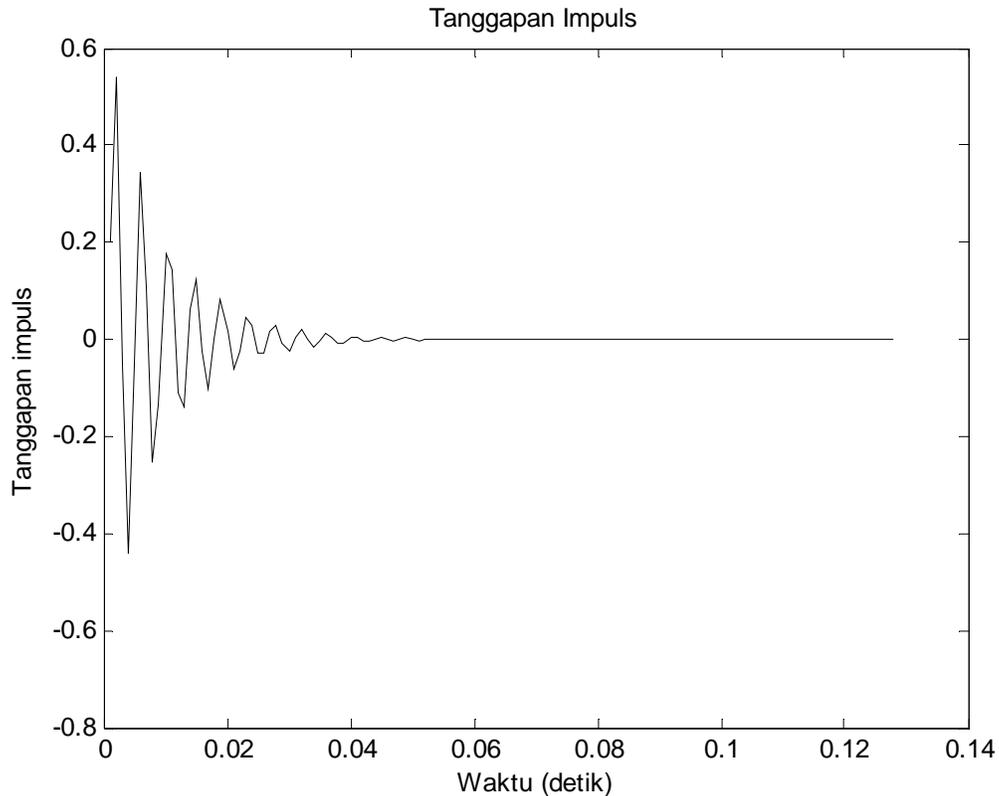
% Menghitung tanggapan impulse
x = [1, zeros(1,N-1)]        % Membangkitkan fungsi impuls
y = filter(b,a,x);           % Menerapkan pers (6)

figure;
t = (1:N)/fs;
plot(t(1:128),y(1:128), 'k'); % Plot tanggapan impuls
xlabel('Waktu (detik)');
ylabel('Tanggapan impuls');

```



(a)



(b)

Gambar 2. (a) Karakteristik frekuensi dari filter pada contoh 1  
(b) Tanggapan impuls

### Filter FIR (Finite Impulse Response)

Filter FIR mempunyai fungsi transfer yang hanya punya pembilang saja yaitu  $H(z) = B(z)$ . Hal ini menyebabkan tanggapan impulsnya berhingga sehingga disebut FIR (*Finite Impulse Response* = Tanggapan impuls berhingga). Kelebihan dari filter FIR adalah bahwa filter ini selalu stabil dan mempunyai geseran fase yang linear, namun kurang efisien dalam hal waktu dan memori (jika ditinjau dari pemakaian komputer).

Persamaan umum filter FIR dapat dinyatakan sbb:

$$y(k) = \sum_{n=1}^L b(n) x(k - n) \quad (8)$$

Dengan  $b(n)$  adalah fungsi koefisien (disebut juga fungsi bobot) dengan panjang  $L$ ,  $x(n)$  adalah input, dan  $y(k)$  adalah output. Persamaan tsb identik dengan

persamaan konvolusi dengan tanggapan impuls  $h(n)$  diganti dengan  $b(n)$ . Dengan demikian filter FIR dapat diimplementasikan dalam Matlab dengan menggunakan fungsi “filter” atau “conv”. Persamaan (8) juga dapat diartikan bahwa koefisien filter FIR adalah sama dengan tanggapan impuls filter. Oleh karena tanggapan frekuensi sebuah proses yang mempunyai tanggapan impuls  $h(n)$  adalah alihragam Fourier dari  $h(n)$ , maka tanggapan impuls FIR dengan koefisien  $b(n)$  adalah alihragam Fourier dari  $b(n)$  itu sendiri,

$$X(m) = \sum_{n=0}^{N-1} b(n) e^{-j2\pi mn/N} \quad (9)$$

Ini adalah kasus khusus jika penyebut dalam persamaan (5) adalah sama dengan satu.

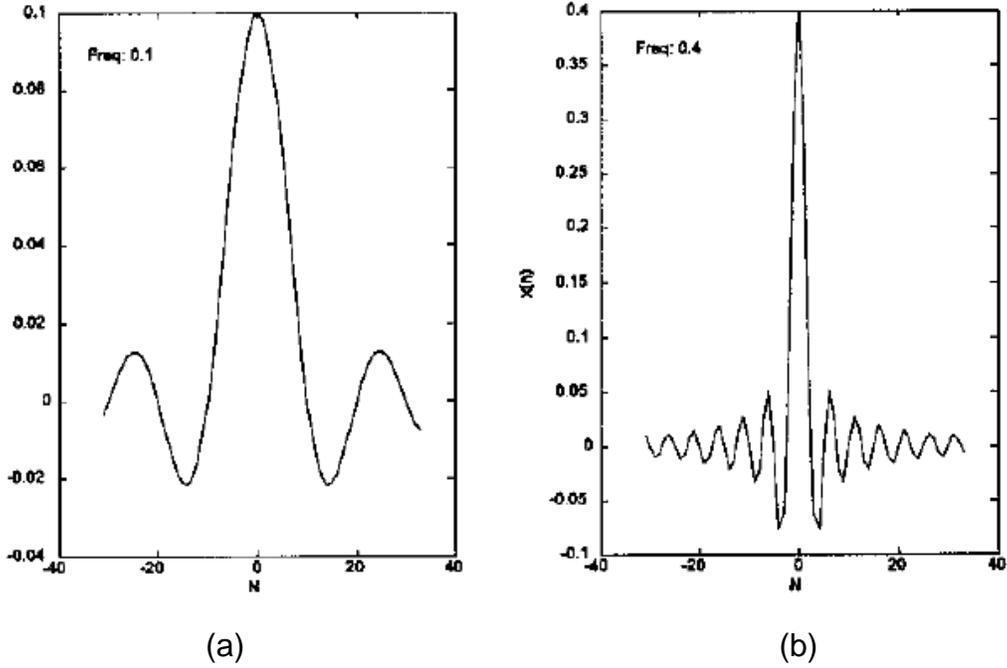
Operasi inversnya, yaitu bekerja dari tanggapan frekuensi yang dikehendaki dan kemudian mencari fungsi koefisien  $b(n)$  disebut dengan **perancangan filter**. Karena tanggapan frekuensi adalah alihragam Fourier dari koefisien filter, maka fungsi koefisien dapat dicari dengan invers alihragam Fouriernya.

### Perancangan Filter

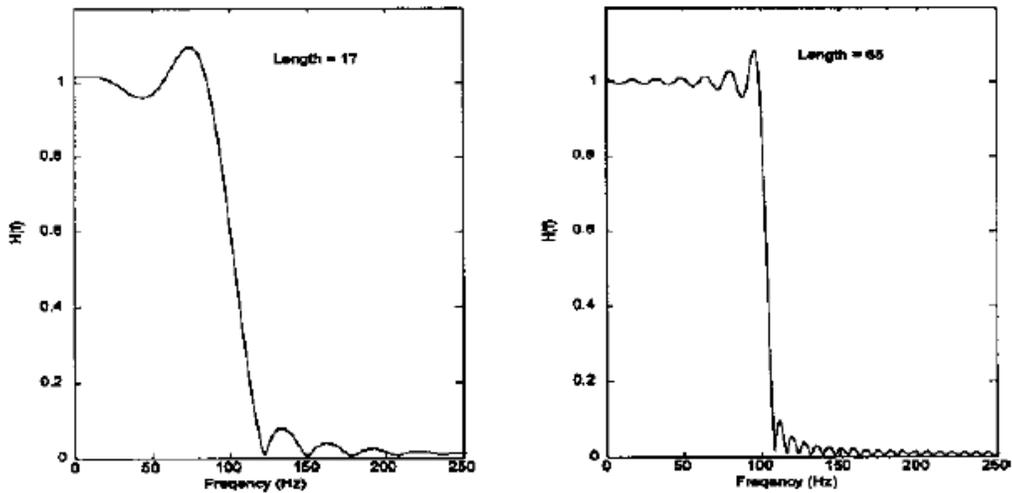
Filter *lowpass* ideal dapat dinyatakan sebagai jendela kotak dalam domain frekuensi. Invers alihragam Fourier untuk memperoleh  $b(n)$  dapat dinyatakan sebagai,

$$b(n) = \frac{\sin[2\pi f_c T_s(n - L/2)]}{\pi(n - L/2)} \quad (10)$$

Dengan  $f_c$  adalah frekuensi *cutoff*,  $T_s$  adalah interval pencuplikan dalam detik, dan  $L$  adalah panjang filter. Argumen  $n - L/2$  digunakan untuk membuat fungsi koefisiennya simetris sehingga karakteristik fasenya linear. Gambar 3 memperlihatkan contoh fungsi koefisien yang simetris untuk dua frekuensi *cutoff* yang berbeda. Pada Gambar 4 diperlihatkan karakteristik frekuensi filter FIR berdasarkan fungsi koefisien pada Gambar 3.

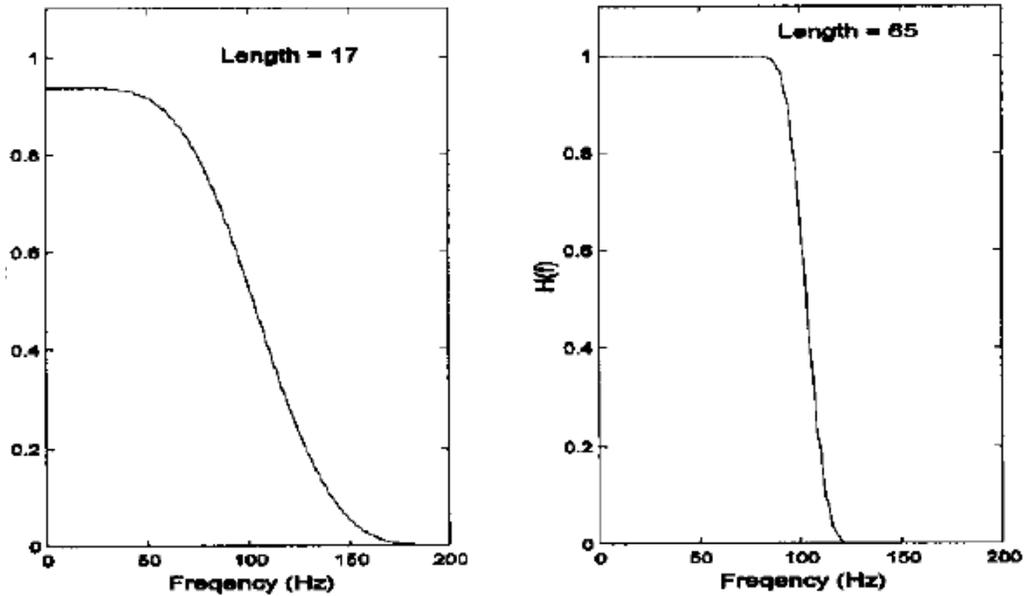


Gambar 3. Fungsi koefisien yang simetris (a) Frekuensi cutoff =  $0.1 f_s / 2$  Hz  
 (b) Frekuensi cutoff =  $0.4 f_s / 2$  Hz



Gambar 4. Karakteristik frekuensi filter FIR berdasarkan fungsi koefisien persamaan (10). Fungsi koefisien dipotong pada koefisien ke-17 dan ke-65. Frekuensi cutoff lowpass sama dengan 100 Hz.

Pada Gambar 5 diperlihatkan karakteristik frekuensi filter FIR yang sama dengan yang digunakan pada Gambar (4) namun dengan jendela Hamming.



Gambar 5. Karakteristik frekuensi filter FIR yang sama dengan yang digunakan pada Gambar (4) namun dengan jendela Hamming.

Berikut adalah persamaan umum untuk koefisien filter *highpass*, *bandpass*, dan *bandstop*.

*Highpass*:

$$b(n) = \frac{\sin[\pi(n - L/2)]}{\pi(n - L/2)} - \frac{\sin[2\pi f_c T_s(n - L/2)]}{\pi(n - L/2)} \quad (11)$$

*Bandpass*:

$$b(n) = \frac{\sin[2\pi f_H T(n - L/2)]}{\pi(n - L/2)} - \frac{\sin[2\pi f_L T_s(n - L/2)]}{\pi(n - L/2)} \quad (12)$$

*Bandstop*:

$$b(n) = \frac{\sin[2\pi f_L T(n - L/2)]}{\pi(n - L/2)} + \frac{\sin[\pi(n - L/2)]}{\pi(n - L/2)} - \frac{\sin[2\pi f_H T_s(n - L/2)]}{\pi(n - L/2)} \quad (13)$$

Salah satu yang menarik dari filter FIR adalah filter yang digunakan untuk membangun turunan (derivatif) dari sebuah sinyal karena turunan sebuah sinyal seringkali sangat penting dalam analisis *biosignals*, misalnya sinyal ECG.

## Operasi Derivatif: Algoritma Two-Point Central Difference

Diferensiasi secara digital didefinisikan sebagai  $\Delta x/\Delta t$  dan dapat diimplementasikan dengan cara mengambil perbedaan antara dua titik yang berurutan, mengalikannya dengan  $1/T_s$ , dan mengulangi operasi ini untuk semua titik pada sinyal. Dalam konteks filter FIR, hal ini identik dengan filter dua koefisien yaitu  $[-1, +1]/T_s$  dan pendekatan inilah yang digunakan dalam Matlab pada fungsi "derv".

Algoritma *two-point central difference* menggunakan dua koefisien yang sama besar namun berlawanan tanda yang dipisahkan sejauh L titik, dan dapat dinyatakan dengan persamaan berikut.

$$y(n) = \frac{x(n + L) - x(n - L)}{2LT_s} \quad (14)$$

Dengan L adalah faktor lompatan (*skip factor*) yang akan mempengaruhi lebar-bidang efektif dan  $T_s$  adalah interval pencuplikan. Koefisien filter untuk algoritma *two-point central difference* menjadi,

$$h(n) = \begin{cases} -0.5/L & n = -L \\ 0.5/L & n = +L \\ 0 & n \neq L \end{cases} \quad (15)$$

### Contoh 2

Bangkitkan fungsi koefisien untuk untuk algoritma *two-point central difference* dan gambarkan tanggapan frekuensinya.

### Penyelesaian

```
% Contoh 2 Bab Filter Digital
% Plot tanggapan frekuensi algoritma two-point central difference

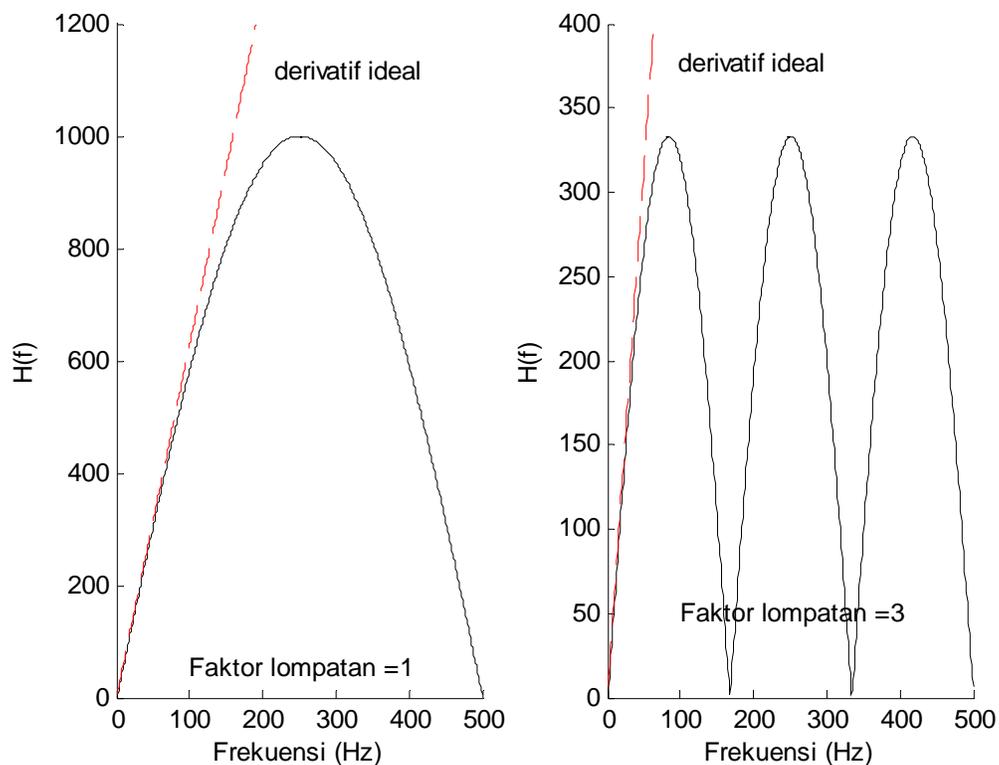
clear all; close all; clc;
Ts = .001; % Ts = 1 milidetik
N = 1000; % Dalam 1 detik data, N = 1000
Ln = [1, 3]; % Menentukan dua faktor lompatan yg berbeda
for i = 1:2 % Ulangi untuk tiap faktor lompatan
    L = Ln(i);
    bn = zeros((2*L)+1,1) % Set b(n), inisialisasi dg nol
    bn(1,1) = -1/(2*L*Ts) % Koef. negatif pada b(1)
```

```

bn((2*L)+1,1)= 1/(2*L*Ts)      % Koef. positif pada b(2L+1)
H = abs(fft(bn,N))             % Mencari tanggapan frekuensi

subplot(1,2,i);
hold on;
plot (H(1:500), 'k');
axis([0 500 0 max(H) + .2*max(H)]);
text(100,max(H)+ 50, ['Faktor lompatan =',Num2str(L)]);
xlabel('Frekuensi (Hz)');
ylabel('H(f)');
y = (1:500)*2*pi;
plot(y, '--r');
end

```



Gambar 6. Tanggapan frekuensi filter FIR untuk algoritma *two-point central difference* dengan faktor lompatan 1 dan 3.

### Contoh 3

Bangun dan aplikasikan sebuah filter *bandpass* pada sebuah sinyal nyata.

## Penyelesaian

Analisis spektrum akan dilakukan pada sinyal asli dan sinyal yang telah difilter dengan filter FIR *bandpass*. Akan digunakan fungsi FFT yang disediakan Matlab dan untuk proses pemfilteran akan digunakan fungsi conv. Filter *bandpass* akan mentransmisikan energi sinyal pada pita tengah (*band*) dan akan menekan komponen frekuensi rendah dan frekuensi tinggi. Pada contoh ini akan digunakan sinyal suara andi.mat dengan bunyi “berhenti”.

```
% Contoh 3 Bab Filter Digital
% Aplikasi Filter FIR Bandpass dengan jendela kotak

close all; clear all;
clc;
N = 6400;           % Jumlah titik data
fs = 8000;         % Frekuensi pencuplikan
load andi;         % Mengambil data
wh = .3*pi;        % Set frek. cutoff filter bandpass
wl = .1*pi;
L = 128;           % Jumlah koef = 128
for i = 1:L+1      % Membangkitkan fungsi koef bandpass
    n = i-L/2;     % dan membuatnya simetris
    if n == 0
        bn(i) = wh/pi-wl/pi;
    else
        bn(i) = (sin(wh*n))/(pi*n)-(sin(wl*n))/(pi*n) % b(n) utk bandpass
    end           % pers (12)
end

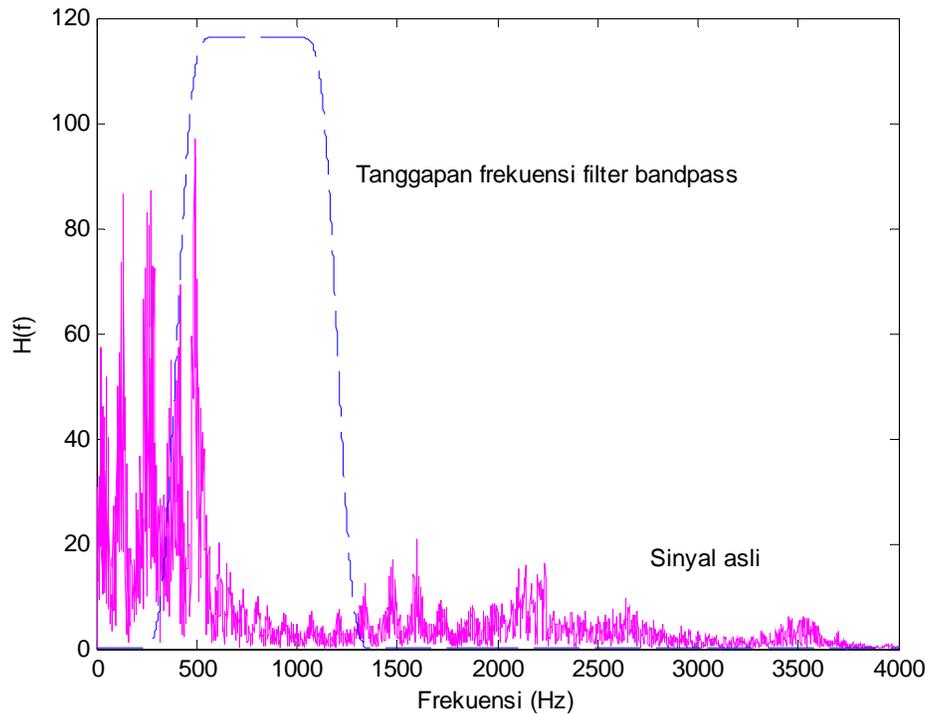
bn = bn.*blackman(L+1)'; % Menerapkan jendela Blackman pd
                             % koef. filter
H_data = abs(fft(berhenti)); % FFT sinyal asli
frek = (1:N/2)*fs/N;        % Vektor frekuensi utk plot
plot(frek, H_data(1:N/2), 'm'); % Plot FFT sinyal asli s.d. fs/2
hold on;
H = abs(fft(bn, N));        % Mencari tangg. frek.dari filter
H = H*1.2*(max(H_data)/max(H)); % Menskala H(z) utk perbandingan
plot(frek, H(1:N/2), '--b'); % Plot tanggapan frekuensi
xlabel('Frekuensi (Hz)');
ylabel('H(f)');

y = conv(berhenti, bn);    % Memfilter sinyal menggu. konvolusi
figure
t = (1:N)/fs;              % Vektor waktu utk plot
subplot(2,1,1);
plot(t(1:N),berhenti(1:N), 'k'); % Plot sinyal asli
xlabel('Waktu (detik)');
```

```

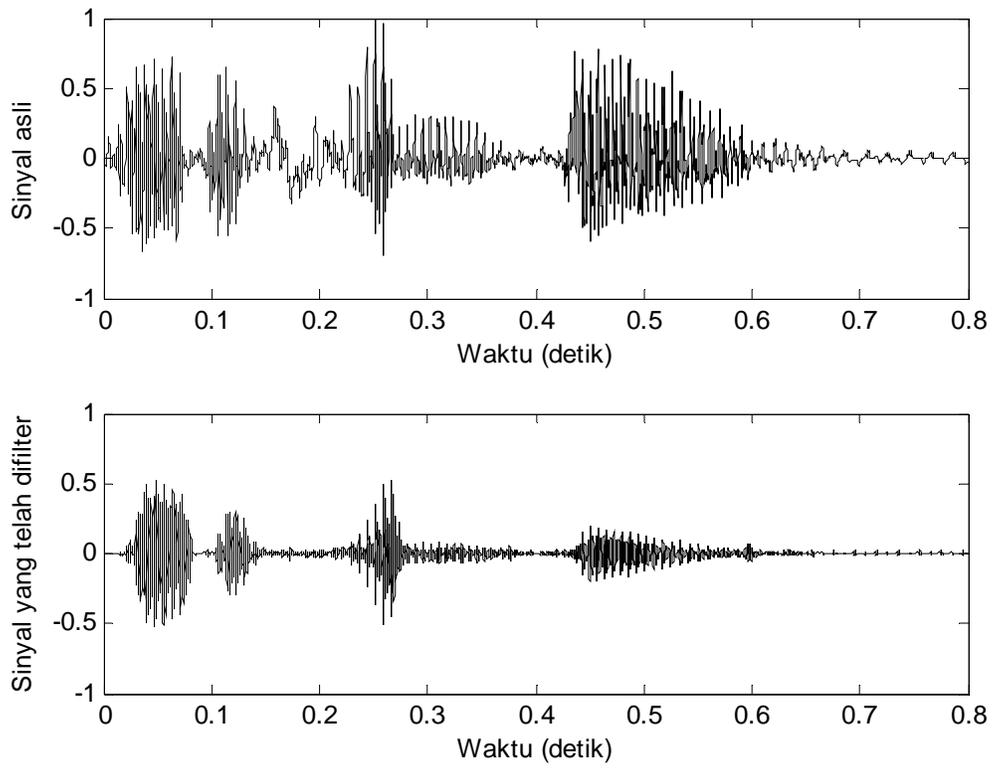
ylabel('Sinyal asli');
subplot(2,1,2);
plot(t(1:N),y(1:N),'k');           % Plot sinyal yg telah difilter
xlabel('Waktu (detik)');
ylabel('Sinyal yang telah difilter');

```



Gambar 7. Tanggapan frekuensi filter FIR *bandpass* (garis putus-putus warna biru) dan alihragam Fourier dari sinyal asli

Pada Gambar 7 diperlihatkan tanggapan frekuensi filter FIR *bandpass* dengan frekuensi *cutoff* rendah dan tinggi masing-masing adalah  $0,1\pi$  dan  $0,3\pi$  radian/sampel atau  $0,1 f_s / 2$  Hz dan  $0,3 f_s / 2$  Hz. Dengan  $f_s$  sebesar 8 kHz maka ini berarti bersesuaian dengan frekuensi 400 hingga 1200 Hz. Pada Gambar 8 diperlihatkan sinyal asli dan sinyal yang telah difilter dengan filter FIR *bandpass* yang telah dibangun. Sinyal yang telah difilter hanya memuat energi sinyal yang mempunyai frekuensi antara 400 hingga 1200 Hz saja.



Gambar 8. Sinyal asli dan sinyal yang telah difilter dengan filter FIR *bandpass*