

# Tugas Desain Filter FIR dan IIR

Nama : Fashalli Giovi Bilhaq

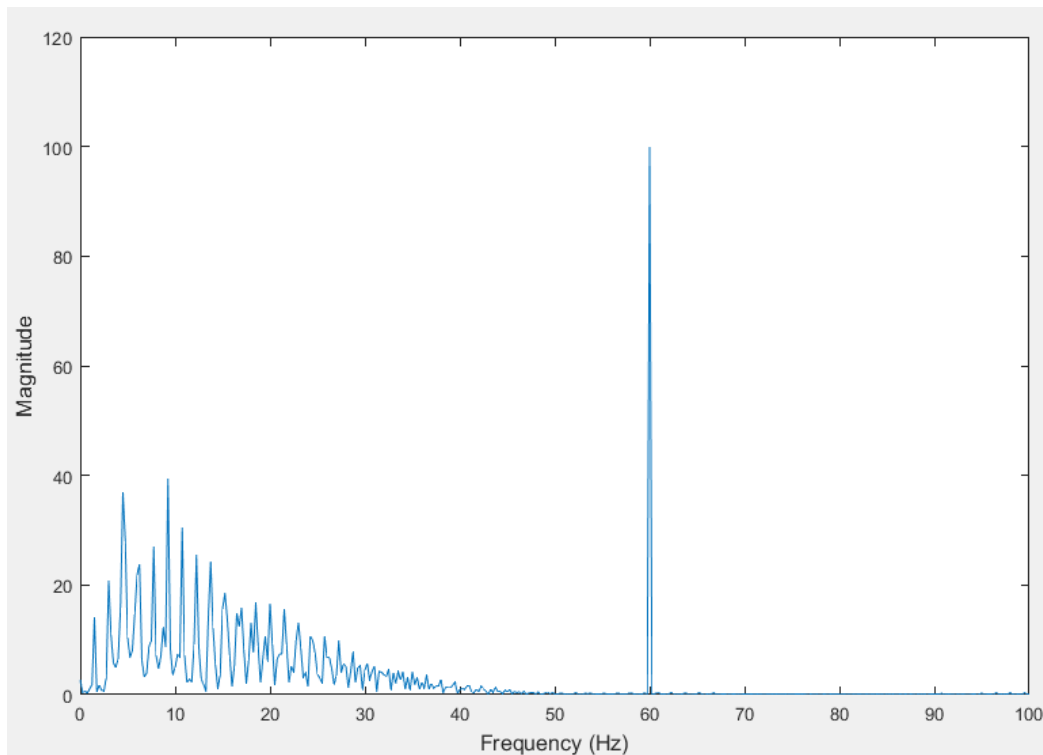
NIM : 081711733026

Digunakan Sinyal dari file ECG60.mat dengan **Sampling Frequency 250 Hz** dan **Noise 60 Hz**. Setelah dilihat pola sinyal yang dimiliki dari ECG60, hanya terdapat data pada range frekuensi 0 - 50 Hz dan Noise pada 60 Hz, sementara di atas 60 Hz tidak terdapat data. Oleh karena itu, dapat digunakan Low Pass Filter maupun Bandstop Filter.

## Analisis Karakter Sinyal:

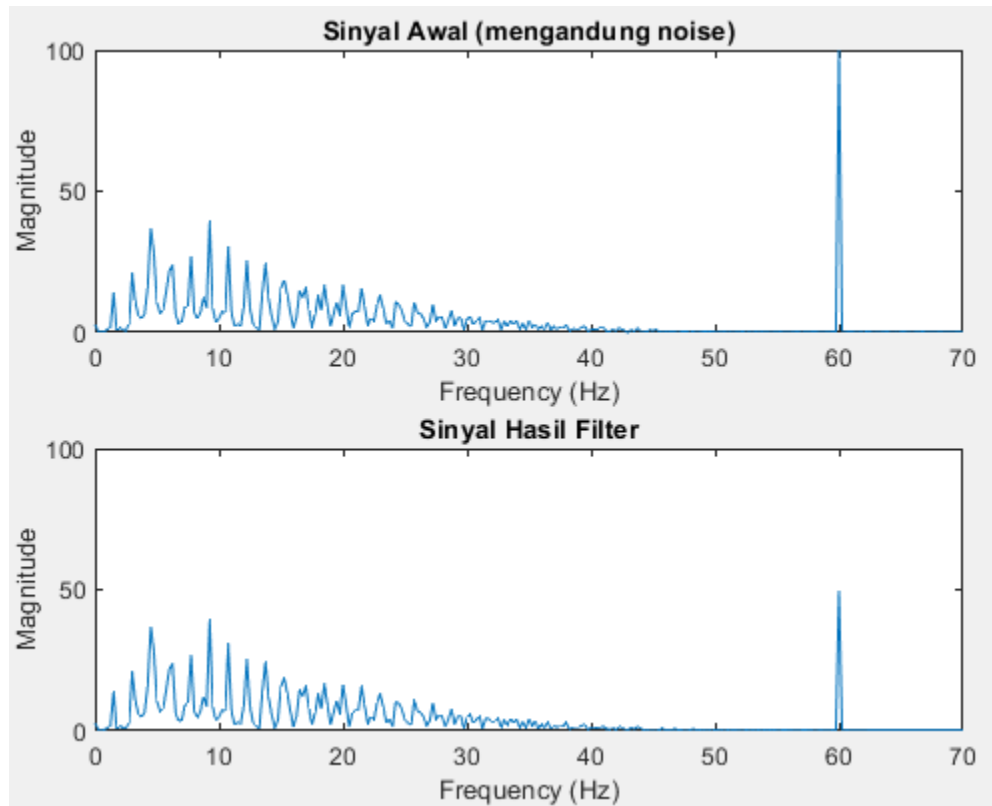
```
clear all; clc;
load ECG60.mat;
N=length(x);
fs=250; %sampling freq dr sinyal
t=(1:N)/fs; %vektor waktu
bins=[0:N-1]*(fs/N); %freq vector untuk plot di fft

figure (1)
y = abs(fft(x)); %fft dari sinyal original
plot(bins, y); %plot hasil fft
ylabel('Magnitude', 'FontSize', 12);
xlabel('Frequency (Hz)', 'FontSize', 12);
xlim([0 100]);
```



## Finite Impulse Response

Digunakan Low Pass Filter dengan frekuensi cut-off (sesuai dengan noise artifisial yang dimiliki sinyal) sebesar 60 Hz, dan 100 koefisien. Saat diimplementasikan pada MATLAB, frekuensi cut-off perlu dinormalisasi dengan cara membaginya dengan frekuensi sampling. Namun saat digunakan nilai 60/250 didapat hasil filtering yang tidak efektif. Masih terdapat noise pada frekuensi 60 Hz.



Maka nilai frekuensi cut-off normalisasi diganti menjadi  $f_c/(f_s*2)$  dan didapatkan hasil yang sangat bersih dari noise 60 Hz.

Untuk meminimalisir ripple yang muncul, digunakan metode *windowing blackmann*, dengan pertimbangan adanya frekuensi transisi pada filtering ini masih dapat ditoleransi.

### Spesifikasi Low Pass Filter - FIR:

- Sampling Frequency: 250 Hz
- Cut-Off Frequency Normalized:  $60/(f_s*2)$
- Filter Length (Number of Coefficient): 100
- Windowing Method: Blackmann

## Script Program:

```
close all; clear all; clc;
load ECG60.mat;
N = length(x);           % Get data length
fs = 250;                 % Sample frequency
t = (1:N)/fs;             % Time vector for plotting
L = 100;                  % Filter lengths
fc = 60/(fs*2);           % Cutoff frequency: 60 Hz, Normalized

%Penghitungan Koefisien untuk Finite Impulse Response LPF
for k = 1:L               % Generate sin(n)/n function Make symmetrical
    n = k-(L-1)/2;         % n = k - L1/2 where L1 even
    if n == 0
        b(k) = 2*fc;       % Case where denominator is zero.
    else
        b(k) = (sin(2*pi*fc*n))/(pi*n); %Filter impulse response
    end
end

b = b.*blackman(L);       % Apply Blackman window
y = conv(x,b,'same');     % Apply filter

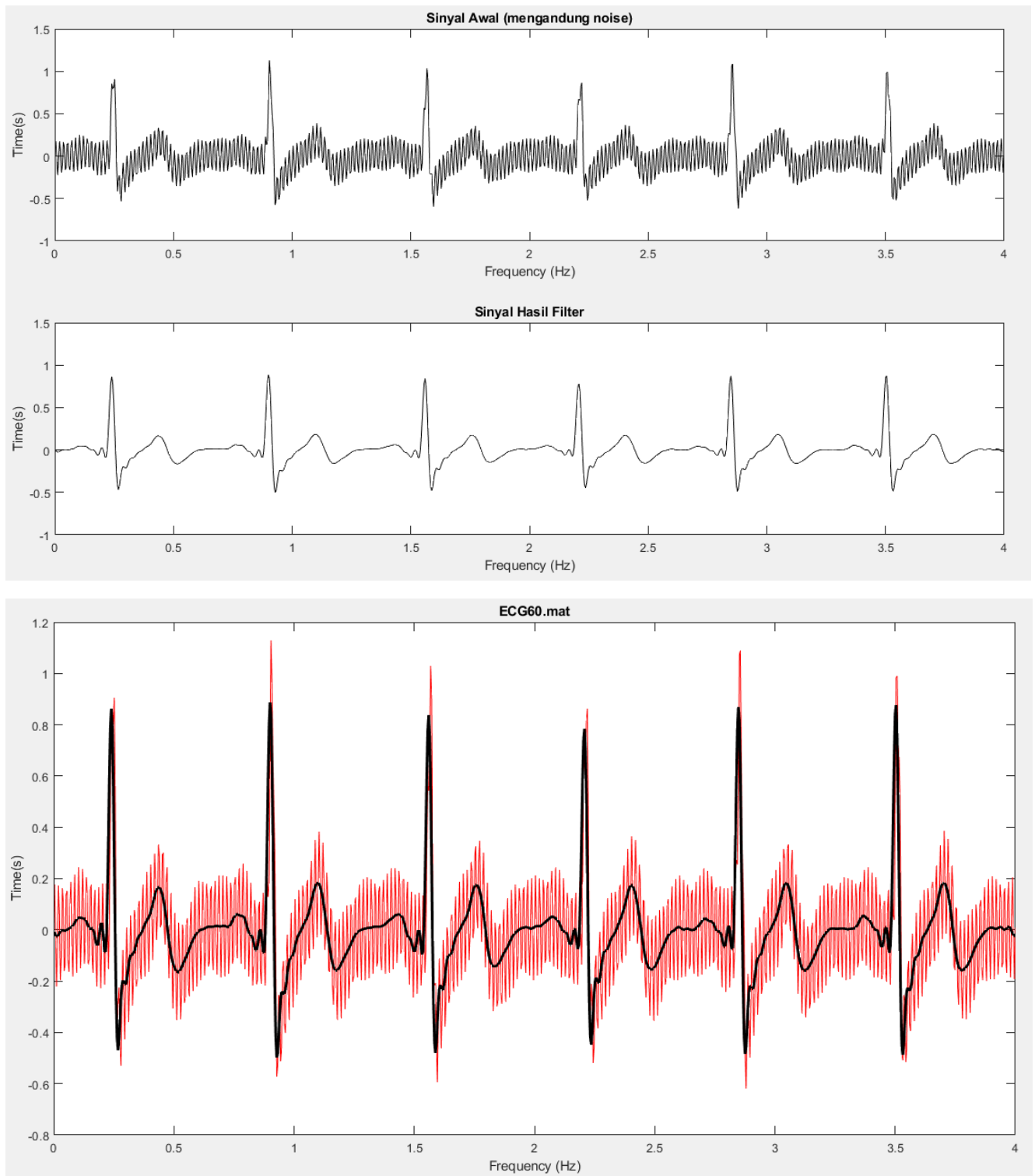
figure(1)
subplot(2,1,1)
plot(t,x,'k');
title('Sinyal Awal (mengandung noise)');
ylabel('Time(s)');
xlabel('Frequency (Hz)');
ylim([-1 1.5]);
subplot(2,1,2)
plot(t,y,'k');
title('Sinyal Hasil Filter');
ylabel('Time(s)');
xlabel('Frequency (Hz)');
ylim([-1 1.5]);

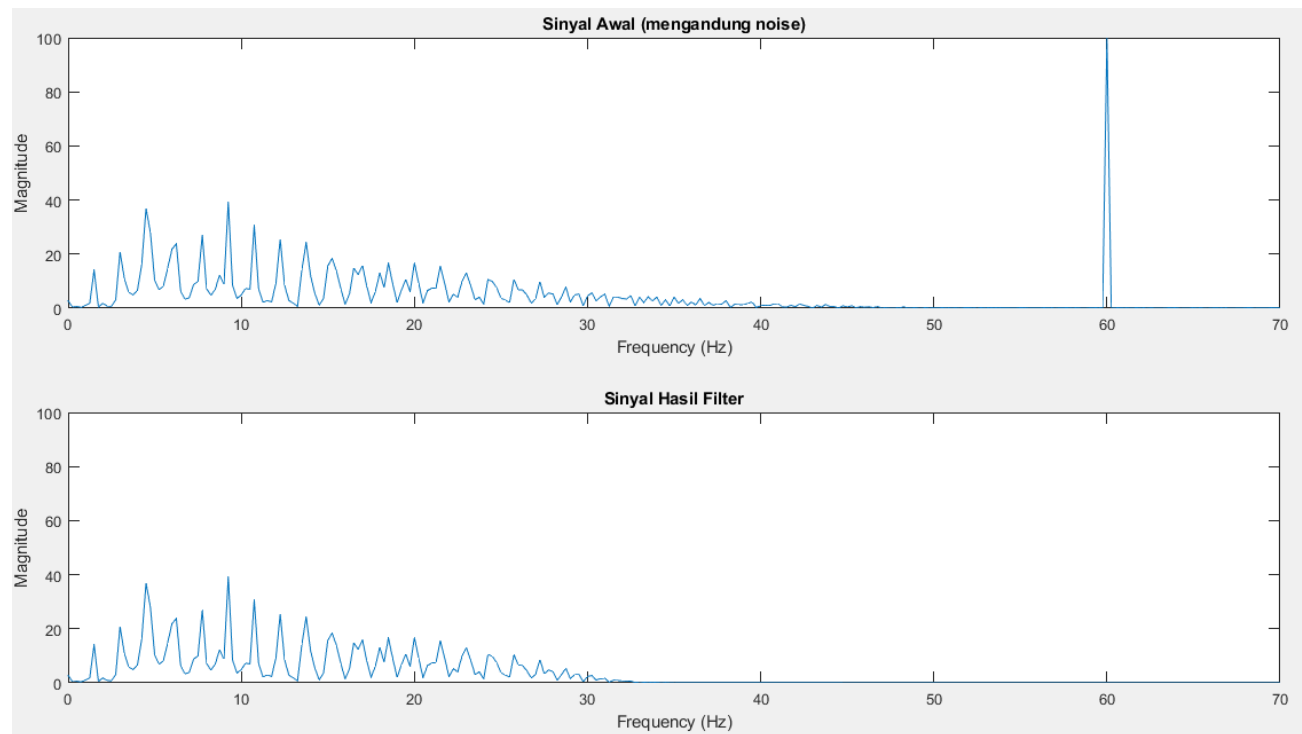
figure(2)
plot(t,x,'r'); hold on;   % Plot original signal
plot(t,y,'k','LineWidth',2); % Plot filtered signal
title('ECG60.mat');
ylabel('Time(s)');xlabel('Frequency (Hz)');

%Cek magnitude spectrum
bins=[0:N-1]*(fs/N);      %freq untuk plotting

figure(3)
subplot(2,1,1)
y2 = abs(fft(x));          %fft sinyal original
plot(bins, y2)
ylim([0 100])
xlim([0 70])
title('Sinyal Awal (mengandung noise)');
ylabel('Magnitude');
xlabel('Frequency (Hz)');
subplot(2,1,2)
a=abs(fft(y));             %fft sinyal hasil filter
plot(bins, a)
ylim([0 100])
xlim([0 70])
title('Sinyal Hasil Filter');
ylabel('Magnitude');xlabel('Frequency (Hz)');
```

## Hasil Filtering:

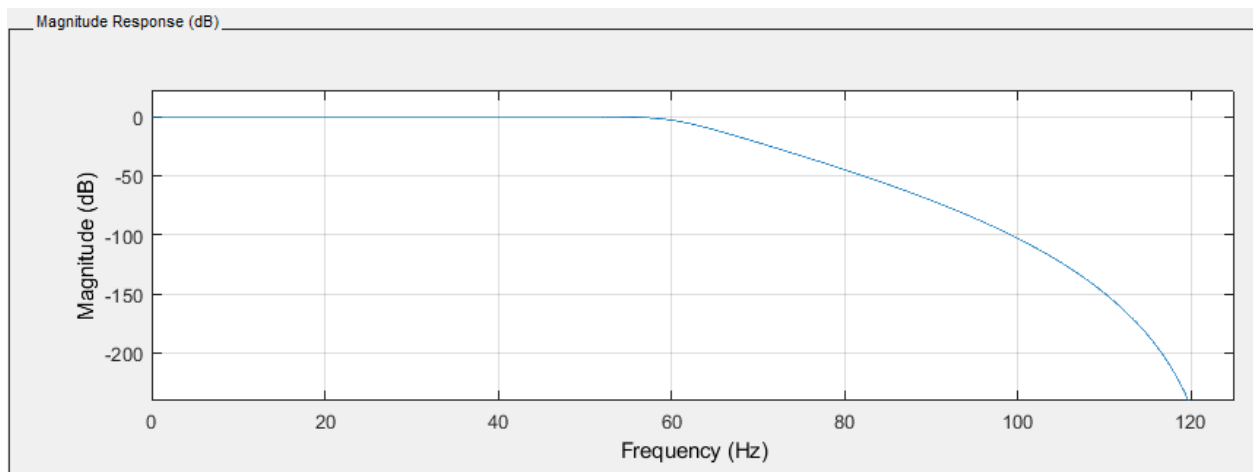




## Infinite Impulse Response

Pada pembuatan desain filter menggunakan metode Infinite Impulse Response, saya menggunakan bantuan toolbox Filter Designer pada MATLAB 2018b untuk menentukan spesifikasi filter yang sesuai dengan kebutuhan dari karakteristik sinyal.

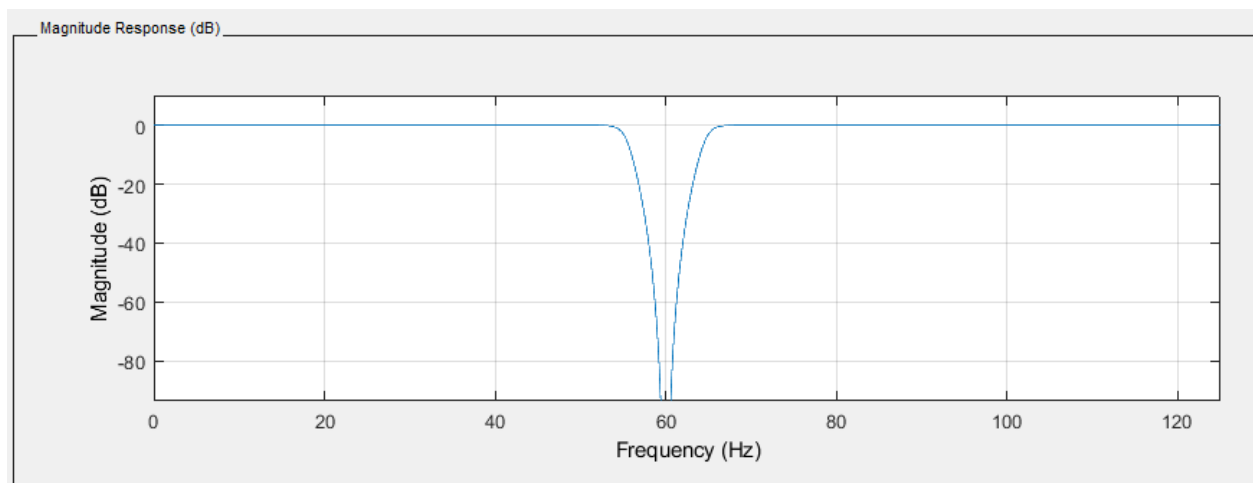
Jika menggunakan pendekatan yang sama dengan FIR sebelumnya yaitu menggunakan Low Pass Filter dengan percobaan pertama menggunakan model *Butterworth* orde 10, didapatkan respon frekuensi sebagai berikut:



Terdapat frekuensi transisi yang terlalu besar.

Dapat digunakan pendekatan dengan metode lain seperti *Chebyshev* maupun *Elliptic*, namun saya memilih *Butterworth* dengan tujuan untuk menghindari ripple yang dapat terjadi.

Maka digunakan model *Butterworth* dengan Bandstop Filter, dengan orde 2, dan transition band 10 Hz (dari 55-65 Hz) didapatkan hasil sebagai berikut:



Setelah filter dilakukan filter pada sinyal, sinyal noise 60 Hz telah hilang, namun masih terjadi atenuasi (terdapat *ripple*) pada sinyal di waktu 0 – 0.2 sekon.

### Spesifikasi Band-Stop Butterworth IIR Filter:

- Sampling Frequency: 250 Hz
- Cut-Off Frequency Normalized:  $60/(fs*2)$
- Transition Frequency Band: 55 – 65

### Script Program:

```
close all; clear all; clc;
load ECG60.mat;
N = length(x); % Get data length
fs = 250; % Sample frequency
t = (1:N)/fs; % Time vector for plotting
fc = 60/(fs*2); % Cutoff frequency: 60 Hz, Normalized
fl = 55/(fs/2);
fh = 65/(fs/2);
rs = 50;
rp = 5;
%Filter IIR
[b,a]=butter(2,[fl fh],'stop');
% [b,a]=cheby2(2,rs,[fl fh],'stop');
% [b,a]=cheby1(2,rp,[fl fh],'stop');
y=filter(b,a,x);

figure(1)
subplot(2,1,1)
plot(t,x,'k');
title('Sinyal Awal (mengandung noise)');
ylabel('Time(s)');
xlabel('Frequency (Hz)');
ylim([-1 1.5]);
subplot(2,1,2)
plot(t,y,'k');
title('Sinyal Hasil Filter');
ylabel('Time(s)');
xlabel('Frequency (Hz)');
ylim([-1 1.5]);

figure(2)
plot(t,x,'r'); hold on; % Plot original signal
plot(t,y,'k','LineWidth',2); % Plot filtered signal
title('ECG60.mat');
ylabel('Time(s)');
xlabel('Frequency (Hz)');

%Cek magnitude spectrum
bins=[0:N-1]*(fs/N); %freq untuk plotting

figure(3)
subplot(2,1,1)
y2 = abs(fft(x)); %fft sinyal original
plot(bins, y2)
ylim([0 100])
xlim([0 70])
title('Sinyal Awal (mengandung noise)');
```

```

        ylabel('Magnitude');
        xlabel('Frequency (Hz)');
        subplot(2,1,2)
        a=abs(fft(y));           %fft sinyal hasil filter
        plot(bins, a)
        ylim([0 100])
        xlim([0 70])
        title('Sinyal Hasil Filter');
        ylabel('Magnitude');
        xlabel('Frequency (Hz)');

```

Hasil Menggunakan Bandstop Filter Butterworth:

