MODUL PRAKTIKUM

# PENGOLAHAN SINYAL DIGITAL

UNIVERSITAS TEKNOLOGI MATARAM

2019

Jl. Kampus Universitas Teknologi Mataram, Kekalik, Mataram

# Buku Modul Praktikum
# Pengolahan Sinyal Digital

SINYAL DAN SISTEM WAKTU DISKRIT

PENGOLAHAN DIGITAL SINYAL WAKTU KONTINYU

STRUKTUR FILTER DIGITAL

DISAIN FILTER DIGITAL

# MODUL 1

# SINYAL DAN SISTEM WAKTU DISKRIT

## 1.1 Tujuan

Memahami konsep deret dan representasinya pada MATLAB

Mempelajari deret-deret dan operasi dasar untuk membentuk deret yang lebih kompleks

Mengerti konsep linearitas, shift-invariance, stabilitas, dan Kausalitas

Menjadi lebih familiar dengan sifat-sifat konvolusi

Mempelajari perhitungan konvolusi menggunakan penjumlahan dan matrix

## 1.2 Peralatan

Program Matlab 2008 keatas

## 1.3 Teori Penunjang

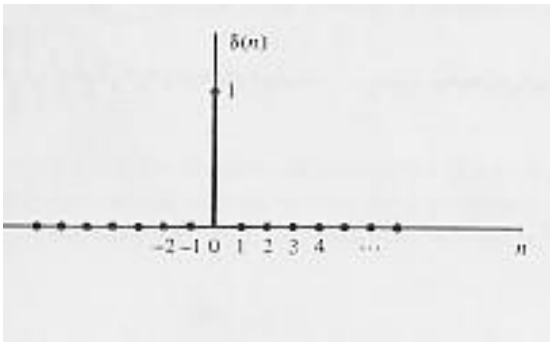
### 1.3.1 Sinyal dan Karakterisitknya

Sinyal waktu diskrit disebut dengan deret dan dinotasikan sebagai berikut:

$$x(n) = \{x(n)\} = \{\ldots, x(-1), x(0), x(1), \ldots\} \quad (1.6)$$

Deret sinyal waktu diskrit dapat berupa deret terbatas maupun tidak terbatas yang terdifinisi pada $N_1 < n < N_2$, dimana $N_1 < N_2$. Dengan durasi deret tersebut adalah $N_2-N_1+1$ sample

Bentuk dasar yang sering digunakan adalah:

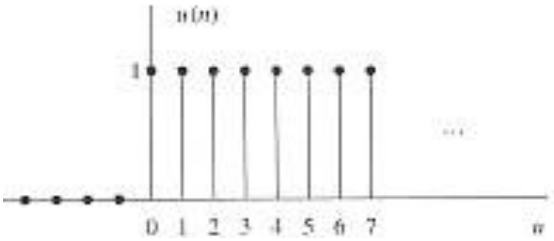



Deret unit sample dinotasikan sebagai d(n) dan didefinisikan sebagai:

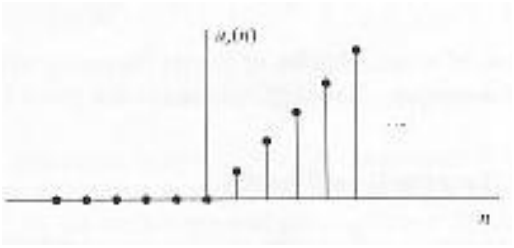
$$(n) \quad \begin{matrix} 1, & ;n \ 0 \\ 0, & ;n \ 0 \end{matrix} \quad (1.2)$$

Sinyal Unit Step dinotasikan sebagai u(n) dan didefinisikan sebagai:

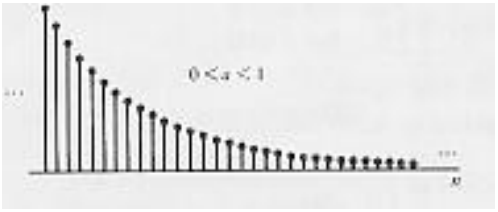
$$u(n) \quad \begin{matrix} 1, \ untuk & n \ 0 \\ 0, \ untuk & n \ 0 \end{matrix} \quad (1.3)$$

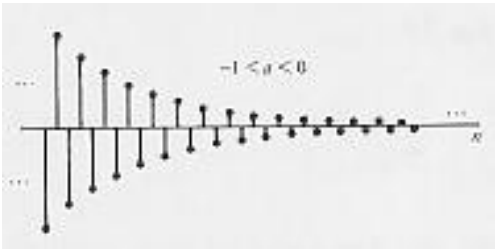



Sinyal Unit Ramp :




Sinyal Eksponensial :




### 1.3.2 Sistem Waktu Diskrit

Sistem waktu diskrit adalah suatu alat atau algoritma yang beroperasi pada pada sinyal waktu diskrit (input), menurut beberapa aturan yang dibuat, untuk menghasilkan sinyal waktu diskrit dengan bentuk lain (output atau respons) sistem tersebut.

Secara umum dinyatakan:

$$y(n) \;\; Tx(n) \qquad (1.4)$$

Salah satu sistem waktu diskrit yang sering digunakan adalah sistem linier tidak berubah terhadap waktu (linier time invariant (LTI) system). Sistem ini mempunyai sifat-sifat sebagai berikut:

Memenuhi sifat superposisi.

Tidak berubah terhadap waktu (time invariant).

Mempunyai respons terhadap deret unit sample yang disebut dengan respons impuls.

Jika input (x(n)) dan sistem (h(n)) adalah deret yg finite maka y(n) merupakan hasil konvolusi dari x(n) dan h(n).

Apabila setiap input yang terbatas menghasilkan output yg terbatas maka sistem disebut dengan stabil BIBO.

Apabila outputnya (y(n)) hanya tergantung dari input n sekarang dan output sebelumnya maka sistem disebut dengan sistem kausal.

Sistem LTI waktu diskrit dapat ditulis/dijelaskan menggunakan persamaan beda koefisien konstanta linier.

### 1.4 Langkah Percobaan

**A Menggambar sinyal waktu diskrit.**

1. Diketahui suatu sinyal $x_1(n)\ (0.9)^n \cos(0{,}2n\ /3)$ 0 < n < 20. Selanjutnya, buatlah program script matlab dan simpan dengan nama "P1_1"

```
clc clear
n1=[0:100];
%x1=((0.9).^n1.*cos(0.2*pi*n1+pi/3));
x2=10*cos(0.008*pi*(n1).^2);
axis([min(n1-1),max(n1-1),-1,1]);
stem(n1,x2)
xlabel('n');ylabel('x2(n)');title(' Deret
x2(n)');
set(gca,'XTickMode','manual','Fontsize',10)
```

2. Jelaskan langkah-langkah pada sript matlab P1-1 diatas.
3. Jalankan program P1_1, dan perhatikan gambar grafik yang dihasilkan. Apakah sinyal di atas adalah sinyal periodic?. Simpanlah gambar yang anda dapatkan tersebut.
4. Modifikasi progarm P1_1 untuk memplot sinyal berikut:
   - $x_2(n)=10\cos(0.008\pi.n^2)$ ; 0< n < 100 ,
   - x3=2n ; 0 < n < 100

   Apakah kedua sinyal ini periodik? Jelaskan.

**B Konvolusi**

1. Diketahui suatu sinyal :
   - $x_4(n)=\{1,2,3,4\}$ ; 0 < n < 3,
   - $x_5(n)=\{3,2,1\}$ ; 0 < n < 2,
   - $x_6(n) = \{2, 2, 1, 2, 3\}$ ; 0 < n < 4.

   Lakukan proses konvolusi untuk $x_4(n)* x_5(n)$, menggunakan program P1_2 berikut:

```
close all clear all
x=input('Enter x:    ')
h=input('Enter h:    ')
m=length(x);
n=length(h);
X=[x,zeros(1,n)];
H=[h,zeros(1,m)];
for i=1:n+m-1
    Y(i)=0;
    for j=1:m
        if(i-j+1>0)
            Y(i)=Y(i)+X(j)*H(i-j+1);
        else
        end
    end
end
Y

stem(Y);
ylabel('Y[n]');
xlabel('----->n');
title('Convolution of Two Signals without
conv function');
```

2. Jalankan program P1_2 dengan ketentuan sebagai berikut:
   - Hitunglah konvolusi $x_4(n)*x_5(n)$ dan $x_5(n)*x_4(n)$, bandingkan hasilnya. Memenuhi sifat konvolusi apakah ini? Jelaskan
   - Hitunglah konvolusi $(x_4(n)*x_5(n))*x_6(n)$ dan $x_4(n)*(x_5(n)*x_6(n))$, bandingkan hasilnya. Memenuhi sifat konvolusi apakah ini? Jelaskan
   - Hitunglah konvolusi $(x_4(n)+x_5(n))*x_6(n)$ dan $x_4(n)*(x_5(n)+x_6(n))$, bandingkan hasilnya. Memenuhi sifat konvolusi apakah ini? Jelaskan

# MODUL 2

# PENGOLAHAN DIGITAL SINYAL WAKTU KONTINYU

## 2.1 Tujuan

Mempelajari hubungan dalam domain waktu antara sinyal waktu kontinyu $x_a(t)$ dan sinyal waktu diskrit x[1] yang dibangkitkan oleh sampling periodik $x_a(t)$

Menginvestigasi hubungan antara frekuensi sinyal sinusoidal $x_a(t)$ dengan perioda sampling.

Menginvestigasi hubungan antara Continuous Time Fourier Transform (CTFT) pada sinyal waktu kontinyu band terbatas (limited) dan Discrete Time Fourier Transform (DTFT) dari sinyal diskrit.

Mendisain filter lowpass analog

## 2.2 PERALATAN

Program Matlab 6.1 ke atas

## 2.3 Teori Penunjang

### 2.3.1 Transformasi Sinyal

Asumsikan $g_a(t)$ adalah sinyal waktu kontinyu yang disample secara kontinyu pada t=nT menghasilkan sekuen g[n], yaitu:

$$g[n] = g_a(nT), \qquad -\infty < n < \infty \tag{2.1}$$

Dengan *T* adalah perioda sampling. Kebalikannya dari T disebut dengan frekuensi sampling ($F_T$), yaitu 1/T. Representasi domain frekuensi dari $g_a(t)$ diperoleh dari transformasi Fourier waktu kontinyu $G_a(j\Omega)$, yaitu :

$$G_a(j\Omega) = \int_{-\infty}^{\infty} g_a(t)\, e^{-j\Omega t}\, dt, \tag{2.2}$$

Dimana representasi domain frekuensi dari g[n] diperoleh dengan transformasi Forirer Diskrit $G(e^{j\omega})$,

$$G(e^{j\omega}) = \sum_{n=-\infty}^{\infty} g[n]\, e^{-j\omega n}. \tag{2.3}$$

Relasi antara $G_a(j\Omega)$ dengan $G(e^{j\omega})$, diberikan oleh :

$$G(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} G_a(j\Omega - jk\Omega_T)|_{\Omega=\omega/T} \tag{2.4}$$

$$= \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a(j\frac{\omega}{T} - jk\Omega_T) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a(j\frac{\omega}{T} - j\frac{2\pi k}{T})$$ 2.5

Atau dapat dinyatakan sebagai:

$$G(e^{j\Omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a(j\Omega - jk\Omega_T)$$ 2.6

### 2.3.2 Teorema Sampling

Asumsikan $g_a(t)$ adalah sinyal bandlimited dengan $G_a(j\Omega) = 0$ untuk $|\Omega| > \Omega_m$. Kemudian $g_a(t)$ dihitung dengan mensampleny a pada $g_a(nt)$, n = 0,1,2,3,4,5, ..... jika,

$\Omega_T > \Omega_m$, dengan $\Omega_T = \frac{2\pi}{T}$ 2.7

Dengan mengetahui {g[n]} = {$g_a(nT)$}, kita dapat memulihkan $g_a(t)$ dengan membangkitkan deret impulse $g_p(t)$, yaitu:

$$g_p(t) = g_a(t)\,p(t) = \sum_{n=-\infty}^{\infty} g_a(nT)\delta(t - nT)$$ 2.8

dan melewatkan $g_p(t)$ ke filter lowpass ideal $H_r(j\Omega)$ dengan gain T dan frekuensi cutoff $\Omega_c > \Omega_m$ dan $\Omega_c < \Omega_T - \Omega_m$, sehingga:

$$\Omega_m < \Omega_c < (\Omega_T - \Omega_m).$$ 2.9

Frekuensi tertinggi $\Omega_m$ yang terkandung dalam $g_a(t)$ disebut dengan **Frekuensi Nyquist**, yang dinyatakan sebagai:

$\Omega_T > 2\,\Omega_m$ 2.10

dan $2\,\Omega_m$ disebut dengan **Nyquist rate.** Jika rate sampling lebih besar dari rate Nyquist maka disebut dengan Oversampling, dan sebaliknya disebut dengan Undersampling. Jika rate sampling sama dengan rate Nyquist maka disebut dengan Critical sampling.

### 2.3.3 Proses Filterisasi

Response impulse $h_r(t)$ dari filter lowpass ideal secara sederhana diperoleh dengan inverse transformasi Fourier dari response frekuensinya $H_r(j\Omega)$, yaitu:

$$H_r(j\Omega) = \begin{cases} T, & |\Omega| \le \Omega_c, \\ 0, & |\Omega| > \Omega_c, \end{cases}$$ 2.11

Maka:

$$h_r(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H_r(j\Omega)\,e^{j\Omega t}\,dt$$
$$= \frac{T}{2\pi}\int_{-\Omega_c}^{\Omega_c} e^{j\Omega t}d\Omega = \frac{\sin(\Omega_c t)}{\Omega_T t/2}, \quad -\infty \le t \le \infty$$ 2.12

Dan deretan impulse diperoleh dengan :

$$g_p(t) = \sum_{n=-\infty}^{\infty} g[n]\,\delta(t - nT) \qquad 2.13$$

Selanjutnya, output filter lowpass ideal $\hat{g}_a(t)$ diketahui dengan mengkonvolusi $g_p(t)$ dengan response impulse $h_r(t)$.

$$\hat{g}_a(t) = \sum_{n=-\infty}^{\infty} g[n]\,h_r(t - nT) \qquad 2.14$$

Substitusi persamaan 2.12 ke dalam persamaan 2.14 dan asumsikan $\Omega_c = \Omega_T/2 = \pi/T$, maka akan diperoleh:

$$\hat{g}_a(t) = \sum_{n=-\infty}^{\infty} g[n]\,\frac{\sin[\pi(t - nT)/T]}{\pi(t - nT)/T} \qquad 2.15$$

### 2.3.4 Spesikasi Filter

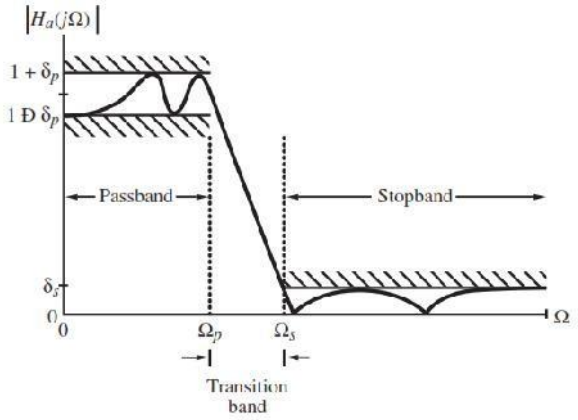
Spesifikasi filter biasanya dinyatakan dalam bentuk respon magnituda.Sebagai contoh, magnituda $|H_a(j\Omega)|$ dari filter laowpass analog ditunjukan pada Gambar 2.1. Dalam *passband*, dinyatakan dengan $0 < \Omega < \Omega_p$, magnitudanya adalah:

$$1 - \delta_p \le |H_a(j\Omega)| \le 1 + \delta_p \quad \text{untuk} \quad |\Omega| \le \Omega_p \qquad 2.16$$

atau dengan kata lain, magnituda mendekati 1 dengan error $\pm \partial_p$. Dalam *stopband* dinyatakan dengan $\Omega_s \le |\Omega| \le \infty$, magnitudanya:

$$|H_a(j\Omega)| \le \delta_s, \qquad \Omega_s \le |\Omega| \le \infty, \qquad 2.17$$

Frekuensi $\Omega_p$ dan $\Omega_s$ masing-masing disebut dengan **passband edge frequency** dan **stopband edge frequency**. Batas toleransi maksimum dalam passband dan stopband $\partial_p$ dan $\partial_s$ disebut dengan **ripples.**




Gambar 2.1 Spesifikasi respon magnituda filter lowpass analog

## 2.4 Proses Sampling dalam Domain Waktu

### 2.4.1 Sampling Sinyal Sinusoidal

Percobaan ini akan menginvestigasi sampling sinyal sinusoidal waktu diskrit $x_a(t)$ di beberapa rate sampling.

1. Buatlah script Matlab berikut dan simpan hasilnya dengan nama "P2_1"

```
% Program P2_1
% Ilustrasi dalam proses sampling domain waktu
clf;
t = 0:0.0005:1;
f = 13;
xa = cos(2*pi*f*t);
subplot(2,1,1)
plot(t,xa);grid
xlabel('Time, msec');ylabel('Amplitude');
title('Continuous-time signal x_{a}(t)');
axis([0 1 -1.2 1.2])
subplot(2,1,2);
T = 0.1;
n = 0:T:1;
xs = cos(2*pi*f*n);
k = 0:length(n)-1;
stem(k,xs); grid
xlabel('Time index n');ylabel('Amplitude');
title('Discrete-time signal x[n]');
axis([0 (length(n)-1) -1.2 1.2])
```

2. Jalankan Program P2_1 untuk menghasilkan sinyal waktu kontinyu dan sinyal versi tersample.
3. Dari Scipt diatas, berapakah frekuensi (Hz) sinyal sinusoidal dan berapakah perioda sampling (detik).
4. Jalan program P2_1 untuk 4 (empat) nilai perioda sampling baru, masing-masing 2 (dua) lebih rendah dan 2 (dua) lainnya lebih tinggi dari perioda sampling di script. Amati hasilnya dan jelaskan .
5. Ulangi program P2_1 dengan merubah frekuensi sinyal menjadi 3 Hz dan 7. Amati dan jelaskan hasil yang diperoleh.

### 2.4.2 Pengaruh Aliasing Dalam Domain Waktu

Pada percobaan ini, kita akan membangkitkan sinyal kontinyu ekivalin $y_a(t)$ dari sinyal diskrit yang dihasilkan oleh program P2_1 untuk menginvestigasi hubungan antara frekuensi sinyal sinusoidal $x_a(t)$ dengan perioda sampling. Untuk menghasilkan sinyal rekontruksi $y_a(t)$, sinyal x[n] dilewatkan melalui filter lowpass menggunakan persamaan :

$$h_r(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H_r(j\Omega)\, e^{j\Omega t}\, dt$$
$$= \frac{T}{2\pi}\int_{-\Omega_c}^{\Omega_c} e^{j\Omega t} d\Omega = \frac{\sin(\Omega_c t)}{\Omega_T t/2}, \quad -\infty \le t \le \infty. \qquad (2....)$$

Langkah Percobaan:

1. Buat script Matlab dan simpan dengan nama 'P2_2'

```
% Program P2_2
% Ilustrasi efek aliasing dalam domain

clf;
T = 0.1;f = 13;
n = (0:T:1)';
xs = cos(2*pi*f*n);
t = linspace(-0.5,1.5,500)';
ya = sinc((1/T)*t(:,ones(size(n))) -
(1/T)*n(:,ones(size(t)))')*xs;
plot(n,xs,'o',t,ya);grid;
xlabel('Time, msec');ylabel('Amplitude');
title('Reconstructed continuous-time signal
y_{a}(t)');
axis([0 1 -1.2 1.2]);
```

2. Jalan program P2_2 untuk membangkitkan sinyal waktu diskrit x[n] dan sinyal kontinyu ekivalennya $y_a(t)$, dan menampilkannya bersama-sama.
3. Berapa range t dan nilai peningkatan waktu dalam script P2_2?. Berapa range t pada gambar/grafik yang dikeluarkan oleh simulasi?. Selanjutnya ubahlah range t, dan jalankan kembali program P2_2. Jelaskan hasil rekonstruksi sinyal yang dihasilkan
4. Kembalikan range sinyal t ke kondisi semula. Selanjutnya, rubahlah frekuensi sinyal sinusoidal menjadi 3 dan 7 Hz. Apakah terdapat perbedaan antara sinyal diskrit ekivalen dengan yang dihasilkan pada langkah 1. Jika tidak, jelaskan.

## 2.5 Effect of Sampling in the Frequency Domain

Percobaan ini akan meneliti hubungan antara Continuous Time Fourier Transform (CTFT) pada sinyal waktu kontinyu band terbatas (limited) dan Discrete Time Fourier Transform (DTFT) dari sinyal diskrit. Dalam hal untuk mengkonversi sinyal waktu kontinyu $x_a(t)$ menjadi sinyal waktu diskrit ekivalen x[n], diperlukan $x_a(t)$ harus band limited dalam domain frekuensi. Untuk mengilustrasikan efek sampling dalam domain frekuensi, percobaan ini menggunakan sinyal waktu kontinyu eksponensial dengan CTFT yang band limited.

**Langkah Percobaan:**

1. Buat script Matlab dan simpan dengan nama 'P2_3'

```
% Program P2_3
% Ilustrasi efek aliasing dalam domain frekuensi
clf;
t = 0:0.005:10;
xa = 2*t.*exp(-t);
subplot(2,2,1)
plot(t,xa);grid
xlabel('Time, msec');ylabel('Amplitude');
title('Continuous-time signal x_{a}(t)');
subplot(2,2,2)
wa = 0:10/511:10;
ha = freqs(2,[1 2 1],wa);
plot(wa/(2*pi),abs(ha));grid;
xlabel('Frequency, kHz');ylabel('Amplitude');
title('|X_{a}(j\Omega)|');
axis([0 5/pi 0 2]);
subplot(2,2,3)
T=1;
n = 0:T:10;
xs = 2*n.*exp(-n);
k = 0:length(n)-1;
stem(k,xs);grid;
xlabel('Time index n');ylabel('Amplitude');
title('Discrete-time signal x[n]');
subplot(2,2,4)
wd = 0:pi/255:pi;
hd = freqz(xs,1,wd);
plot(wd/(T*pi), T*abs(hd));grid;
xlabel('Frequency, kHz');ylabel('Amplitude');
title('|X(e^{j\omega})|');
axis([0 1/T 0 2])
```

2. Jalankan program P2_3 untuk membangkitkan dan mendisplaykan sinyal waktu diskrit dan sinyal kontinyu ekivalennya, dan kaitan dengan transformasi Fourier. Apakah tampak ada efek aliasing?
3. Ulangi jalankan program P2_3 dengan meningkatkan perioda sampling manjadi 1.5. Apakah terjadi efek aliasing?
4. Modifikasi program P2_3 untuk kasus $x_a(t) = e^{-\pi t^2}$ dan ulangi pertanyaan 2 dan 3.

## 2.6 Disain Filter Lowpass Analog

Tahap pertama dalam mendisain filter adalah menentukan orde filter (N) dan frekuaensi cutoff ($\Omega_c$). Parameter ini dihitung menggunakan fungsi Matlab "**buttord**" untuk filter Butterworth, "**cheb1ord**" untuk filter Chebyshev Tipe 1, "**cheb2ord**" untuk tipe 2, dan "**ellipord**" untuk filter elliptic. $\Omega_c$ adalah frekuensi cutoff 3 dB untuk filter Butterworth, passband edge untuk filter Chebyshev Type 1, stopband edge untuk filter Chebyshev Type 2, dan passband edge untuk filter elliptic.

Langkah Percobaan :

1. Buat script Matlab dan simpan dengan nama 'P2_4'

```
% Program P2_4
% Disain filter lowpass analog
clf;
Fp = 3500;Fs = 4500;
Wp = 2*pi*Fp; Ws = 2*pi*Fs;
[N, Wn] = buttord(Wp, Ws, 0.5, 30,'s');
[b,a] = butter(N, Wn, 's');
wa = 0:(3*Ws)/511:3*Ws;
h = freqs(b,a,wa);
plot(wa/(2*pi), 20*log10(abs(h)));grid
xlabel('Frequency, Hz');ylabel('Gain, dB');
title('Gain response');
axis([0 3*Fs -60 5]);
```

2. Perhatikan script diatas,berapakah passband ripple (Rp) dalam dB dan minimum stopband attenuation (Rs) dalam dB. Berapakah frekuensi passband dan stopband edge (Hz)?
3. Jalankan program P2_4 dan perhatikan display grafik yang dihasilkan.Apakah filter yang dirancang sudah memenuhi spesifikasi ?. Berapakah orde filter (N) dan frekuensi cutoff (Hz) dari filter yang telah dirancang?

# MODUL 3

# STRUKTUR FILTER DIGITAL

## 3.1 TUJUAN

Merealisasikan kaskade fungsi tranfer filter FIR

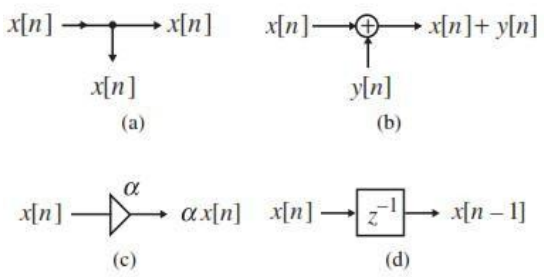
Merealisasikan kaskade fungsi tranfer filter IIR

## 3.2 PERALATAN

.Program Matlab 2008 ke atas

## 3.3 TEORI PENUNJANG

Algoritma komputasi dari filter digital LTI dapat dinyatakan dalam blok-diagram menggunakan blok-blok bangunan dasar seperti unit delay, pengali (multiplier), penjumlah (adder) dan pick-off node).




Gambar 3.1 Blok-blok bangunan dasar: (a) pick-off node, (b) adder, (c) multiplier, dan (d) unit delay

Dua struktur filter digital adalah ekivalen jika memiliki fungsi transfer yang sama. Cara paling mudah untuk membangkitkan struktur yang ekivalen adalah melalui fungsi transpose, yaitu : (i) Membalikan seluruh jalur, (ii) Mengganti pick-off dengan penjumlah (adder) atau sebaliknya, dan (iii) Membalikan node input dan ouput.

Struktur yang koefisien-koefisien pengalinya (multiplier) tepat, koefisiein-koefisien fungsi transfer disebut dengan struktur Direct Form

## 3.4 PERCOBAAN REALISASI FUNGSI TRANSFER FIR

### 3.4.1 Realisasi Kaskade

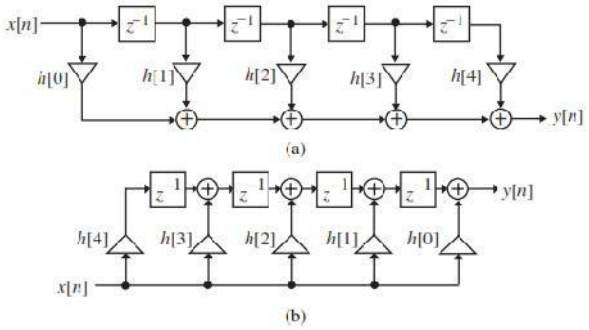
Filter FIR kausal dengan panjang M, dikarakteristikan oleh fungsi transfer H(z):

$$H(z) = \sum_{k=0}^{M-1} h[k]\, z^{-k} \tag{3.1}$$

Dalam domain waktu relasi input-output dinyatakan dengan:

$$y[n] = \sum_{k=0}^{M-1} h[k]\, x[n-k] \tag{3.2}$$

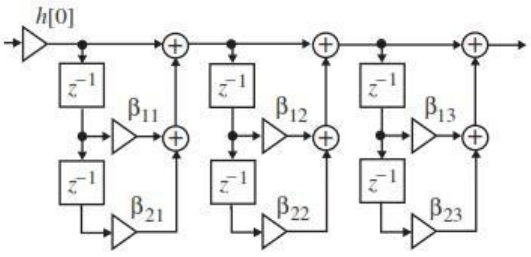
Realiasasi Form Direct dari filter FIR dikembangkan dari persamaan (3.2), ditunjukan pada Gambar 3.2(a) untuk M=5, dan transposenya ditunjukan pada Gambar 3.2(b). Secara umum dalam implementasinya, filter FIR panjan M dikarakteristikan oleh M koefisien, membutuhkan M pengali dan (M-1) penjumlah dua input.




Gambar 3.2 Struktur Direct Form filter FIR

Fungsi transfer FIR orde lebih tinggi dapat direalisasikan dengan kaskade seksi-seksi FIR dengan setiap seksi dikaraketerisitkan oleh fungsi transfer orde perta, atau kedua. Maka, fungsi transfer FIR H(z) dalam bentuk terfaktor, dinyatakan sebagai

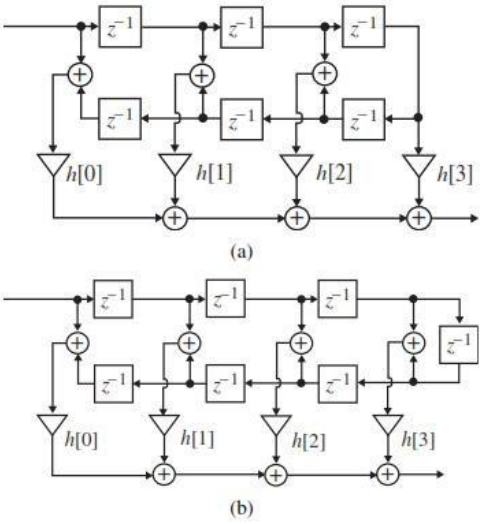
$$H(z) = h[0] \prod_{k} (1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}) \tag{3.3}$$




Gambar 3.3 Struktur bentuk kaskade FIR dengan panjang 7

Fase linier dari filter FIR panjang-M dikarakteristikan oleh kesimetrisan response impulse h[n]=h[M − 1 − n] atau anti-simetris impulse response h[n]=−h[M -1− n]. Sifat simetri dari phase linier filter FIR dapat diekploitasi untuk menurunkan jumlah total pengali menjadi setengah yang dibutuhkan dalam

implementasi fungsi transfer Form Direct. Gambar 3.4 (a)menunjukan realisasi fungsi transfer FIR tipe 1 panjang 7 dengan respon impulse simetris, dan (b) menunjukan realisasi fungsi transfer FIR tipe 1 panjang 8 dengan respon impulse simetris.




Gambar 3.4 Struktur Linear-phase FIR : (a) Tipe 1 and (b) Tipe 2.

### 3.4.2 Langkah Percobaan:

1. Buat script Matlab dan simpan hasilnya dengan nama “p3-1”

```
% Program P3_1
num = input('Numerator coefficient vector = ');
den = input('Denominator coefficient vector = ');
[A, B] = eqtflength(num, den);
[z,p,k] = tf2zp(A, B);
sos = zp2sos(z,p,k)
```

2. Dengan menggunakan Program P3_1, bangunlah sebuah realisasi kaskade untuk fungsi tranfer FIR berikut:

$$H_1(z) = 2 + 10\,z^{-1} + 23\,z^{-2} + 34\,z^{-3} + 31\,z^{-4} + 16\,z^{-5} + 4\,z^{-6}$$

   Sketch blok diagram untuk merealisasikan kaskade

   Apakah $H_1(z)$ adalah fungsi transfer fase linier?

3. Selanjutnya, gunakan Program P3_1 untuk membangun kaskade dengan fungsi tranfer FIR berikut:

$$H_2(z) = 6 + 31\,z^{-1} + 74\,z^{-2} + 102\,z^{-3} + 74\,z^{-4} + 31\,z^{-5} + 6\,z^{-6}$$

   Sketch blok diagram untuk merealisasikan kaskade

   Apakah $H_1(z)$ adalah fungsi transfer fase linier?

## 3.5 PERCOBAAN REALISASI FUNGSI TRANFER IIR

### 3.5.1 Realisasi Kaskade

Filter IIR kausal beroder N dikarakteristikan oleh fungsi transfer H(z)::

$$H(z) = \frac{\sum_{k=0}^{N} p_k\, z^{-k}}{1 + \sum_{k=1}^{N} d_k\, z^{-k}}, \tag{3.4}$$

Dalam domain waktu, relasi input-output filter IIR dinyatakan dengan:

$$y[n] = \sum_{k=0}^{N} p_k\, x[n-k] - \sum_{k=1}^{N} d_k\, y[n-k] \tag{3.5}$$

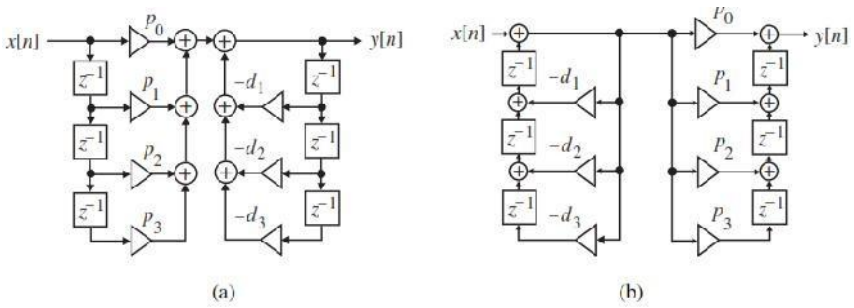
Dengan mendefinisikan variabel sinyal intermediate, w[n],

$$w[n] = \sum_{k=0}^{N} p_k\, x[n-k] \tag{3.6}$$

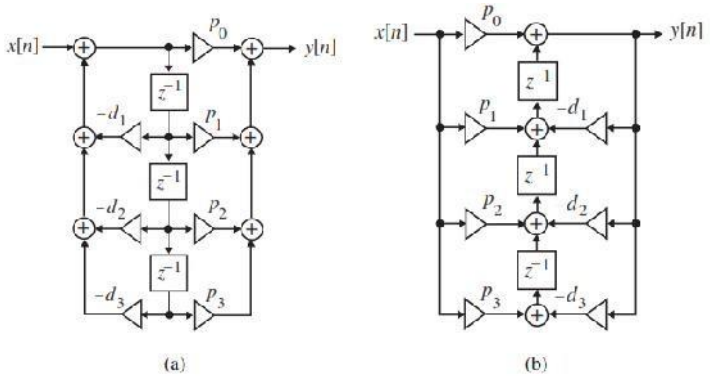
Makan persamaan (3.5) dapat dinyatakan sebagai:

$$y[n] = w[n] - \sum_{k=1}^{N} d_k\, y[n-k] \tag{3.7}$$

Realisasi filter IIR berdasarkan persaman (3.6) dan (3.7) disebut dengan struktur Direct Form I, seperti ditunjukan oleh Gambar 3.5(a) untuk N=3, dan bentuk transposenya ditunujukan pada Gambar 3.5(b). Jumlah total delay yang diperlukan dalam realisasi Direct Form I adalah 2N, dapat diturunkan menjadi N, dengan memanipulasi diagram blok menghasilkan struktur Direct Form II, seperti ditunjukan pada Gambar 3.6 (N=3).




Gambar 3.5 (a) Struktur Direct Form I, (b) Struktur Transpose Direct Form II



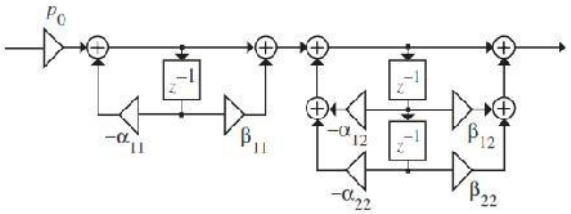

Gambar 3.6 (a) Struktur Direct Form II, (b) Struktur Transpose Direct Form II

Dengan menyatakan polinomial pembilang (numerator) dan penyebut (denominator) dari fungsi transfer H(z) sebagai perkalian dari plinomial-polinomial orde rendah, maka filter digital dapat direalisasikan sebagai kaskade dari seksi-seksi filter orde rendah. Pada kasus ini, H(z) dinyatakan sebagai:

$$H(z) = p_0 \prod_k \left( \frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right) \tag{3.8}$$

Untuk orde pertama, faktor $\alpha_{2k} = \beta_{2k} = 0$. Realisasi yang mungkin dari fungsi transfer orde-3 adalah:

$$H(z) = p_0 \left( \frac{1 + \beta_{11} z^{-1}}{1 + \alpha_{11} z^{-1}} \right) \left( \frac{1 + \beta_{12} z^{-1} + \beta_{22} z^{-2}}{1 + \alpha_{12} z^{-1} + \alpha_{22} z^{-2}} \right) \tag{3.9}$$



Gambar 3.7 Realisasi kaskade fungsi transfer IIR orde-3

Fungsi transfer IIR dapat direalisasikan dalam bentuk Paralel Form I, dan Paralel Form II, yaitu:

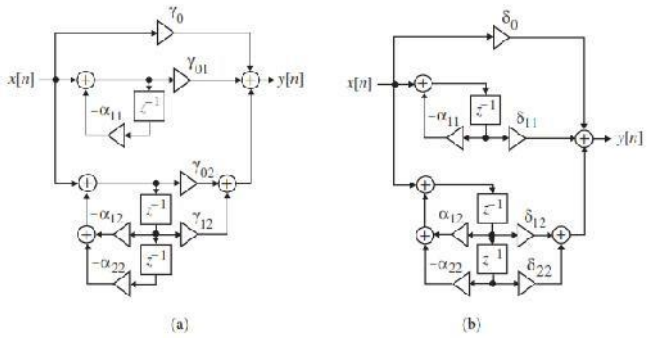
Paralel Form I :

$$H(z) = \gamma_0 + \sum_k \left( \frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right) \tag{3.10}$$

Paralel Form II:

$$H(z) = \delta_0 + \sum_k \left( \frac{\delta_{1k} z^{-1} + \delta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right) \tag{3.11}$$

Realisasi paralel dari fungsi transfer IIR orde-3 ditunjukan pada Gambar 3.8




Gambar 3.8 Realisasi Paralel dari dari fungsi transfer IIR orde-3: (a) Parallel Form I, (b)Parallel Form II.

### 3.5.2 Langkah Percobaan

#### A. Realisasi Kaskade

1. Gunakan **Program P3_1** untuk membangun realisasi kaskade dengan fungsi transfer IIR

$$H_1(z) = \frac{3 + 8\,z^{-1} + 12\,z^{-2} + 7\,z^{-3} + 2\,z^{-4} - 2\,z^{-5}}{16 + 24\,z^{-1} + 24\,z^{-2} + 14\,z^{-3} + 5\,z^{-4} + z^{-5}}$$

   Gambarkan blok diagram dari realisasi kaskade

2. Gunakan **Program P3_1** untuk membangun realisasi kaskade dengan fungsi transfer IIR

$$H_2(z) = \frac{2 + 10\,z^{-1} + 23\,z^{-2} + 34\,z^{-3} + 31\,z^{-4} + 16\,z^{-5} + 4\,z^{-6}}{36 + 78\,z^{-1} + 87\,z^{-2} + 59\,z^{-3} + 26\,z^{-4} + 7\,z^{-5} + z^{-6}}$$

   Gambarkan blok diagram dari realisasi kaskade

#### B. Realisasi Paralel

1. Buat script Matlab dan simpaan hasilnya dengan nama “p3-2”

```
% Program P3_2
% Parallel Form Realizations of an IIR Transfer Function
num = input('Numerator coefficient vector = ');
den = input('Denominator coefficient vector = ');
[r1,p1,k1] = residuez(num,den);
[r2,p2,k2] = residue(num,den);
disp('Parallel Form I')
disp('Residues are');disp(r1);
disp('Poles are at');disp(p1);
disp('Constant value');disp(k1);
disp('Parallel Form II')
disp('Residues are');disp(r2);
disp('Poles are at');disp(p2);
disp('Constant value');disp(k2);
```

2. Gunakan **Program P3_2** untuk membangun realisasi bentuk paralel dengan fungsi transfer IIR

$$H_1(z) = \frac{3 + 8\,z^{-1} + 12\,z^{-2} + 7\,z^{-3} + 2\,z^{-4} - 2\,z^{-5}}{16 + 24\,z^{-1} + 24\,z^{-2} + 14\,z^{-3} + 5\,z^{-4} + z^{-5}}$$

Gambarkan blok diagram dari realisasi Paralel

3. Gunakan **Program P3_2** untuk membangun realisasi bentuk paralel dengan fungsi transfer IIR

$$H_2(z) = \frac{2 + 10\,z^{-1} + 23\,z^{-2} + 34\,z^{-3} + 31\,z^{-4} + 16\,z^{-5} + 4\,z^{-6}}{36 + 78\,z^{-1} + 87\,z^{-2} + 59\,z^{-3} + 26\,z^{-4} + 7\,z^{-5} + z^{-6}}$$

Gambarkan blok diagram dari realisasi Paralel

# MODUL 4

# DISAIN FILTER DIGITAL

## 4.1 TUJUAN

Disain dan implementasi filter digital IIR

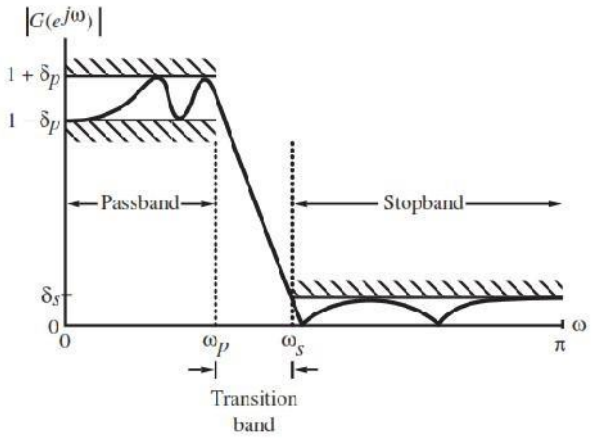
Disain dan implementasi filter digital FIR

## 4.2 PERALATAN

Program Matlab 2008 keatas

## 4.3 TEORI PENUNJANG

Spesifikasi filter biasanya dinyatakan dalam bentuk response magnitudanya. Sebagai contoh, magnituda $|G(e^{j\omega}|$ dari filter lowpass G(z), dinyatakan seperti pada Gambar 4.1.




Gambar 4.1 Tipikal response magnituida untuk filter digital lowpass

Dalam *passband* didefinisikan oleh $0 \leq \omega \leq \omega_p$, diperlukan:

$$1 - \delta_p \leq |G(e^{j\omega})| \leq 1 + \delta_p, \quad \text{for} \quad |\omega| \leq \omega_p \tag{4.1}$$

Dengan kata lain, magnituda mendekati 1 (satu) dengan kesalahan $\pm \delta_p$

Dalam *Stopband*, didefinisikan oleh $\omega_s \leq |\omega| \leq \pi$, diperlukan:

$$|G(e^{j\omega})| \leq \delta_s, \quad \text{for} \quad \omega_s \leq |\omega| \leq \pi. \tag{4.2}$$

Yang menunjukan bahwa magnituda mendekati 1 *satu) dengankesalahan $\delta_s$

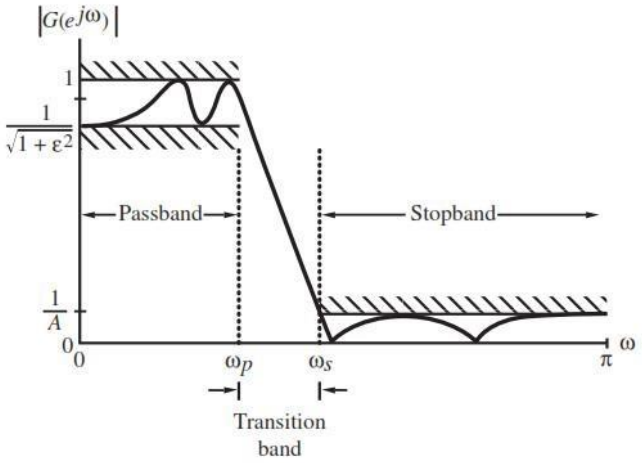
Frekuensi $\omega_p$ dan $\omega_s$, masing-masing disebut dengan *frekuensi tepi passband* dan *frekuensi tepi stopband.* Batas maksimum toleransi dalam passband ($\delta_p$) dan stopband ($\delta_s$), disebut dengan *Ripples*.

Pada banyak aplikasi, spesifikasi filter digital diketahui seperti ditunjukan pada Gambar 4.2. Disini, *Passband* dinyatakan oleh $0 \leq \omega \leq \omega_p$, nilai maksimum dan minimum dari magnituda masing-masing dinyatakan dengan 1 (satu) dan $1/\sqrt{1+e^2}$. *Peak passband ripple* (dB) adalah:

$$R_p = 20 \log_{10} \sqrt{1+\epsilon^2} \text{ dB} \tag{4.3}$$

Maksimum *Ripple* dalam *stopband*, didefinisikan oleh $\omega_s \leq |\omega| \leq \pi$, dinyatakan dengan 1/A, dan maksimum *minimum stopband attenuation* (dB) dinyatakan dengan:

$$R_s = 20 \log_{10} A \text{ dB} \tag{4.4}$$




Gambar 4.2 Spesifikasi respon magnituda ternormalisasi untuk filter digital lowpass

Jika frekuensi tepi passband (Fp) dan stopband (Fs) dari filter digital dinyatakan dalam Hz dengan laju sampling ($F_T$), maka frekuensi angular ternormalisasi dalam radian dinyatakan dengan:

$$\omega_p = \frac{\Omega_p}{F_T} = \frac{2\pi F_p}{F_T} = 2\pi F_p T.$$
$$\omega_s = \frac{\Omega_s}{F_T} = \frac{2\pi F_s}{F_T} = 2\pi F_s T. \tag{4.5}$$

## 4.4 PERCOBAAN DISAIN FILTER LPF IIR

Fungsi transfer yang analog yang biasa digunakan dalam mendisain filter IIR adalah Butterworth, Chebyshev Tipe 1, Chebyshev Tipe 2, dan fungsi transfer elliptic.

### 4.4.1 Estimasi Orde filter IIR

Step pertama dalam proses mendisain filter adalah memilih tipe pendekatan filter yang diterapkan dan kemudian mengestimasi orde fungsi transfer dari spesifikasi filter. Untuk filter **Butterworth,** etimasi orde dapat menggunakan command dari Matlab yaitu:

```
[N,Wn] = buttord(Wp, Ws, Rp, Rs)
```

Untuk filter **Chebyshev Tipe 1,** etimasi orde dapat menggunakan command dari Matlab yaitu:

```
[N, Wn] = cheb1ord(Wp, Ws, Rp, Rs)
```

Untuk filter **Chebyshev Tipe 2**, etimasi orde dapat menggunakan command dari Matlab yaitu:

```
[N, Wn] = cheb2ord(Wp, Ws, Rp, Rs)
```

Untuk filter Elliptic, etimasi orde dapat menggunakan command dari Matlab yaitu

```
[N, Wn] = ellipord(Wp, Ws, Rp, Rs)
```

**Langkah Percobaan**

1. Ketikan command Matlab diatas untuk menghitung order terendah **filter lowpass** IIR pada ke-4 jenis filter, menggunakan spesifikasi filter berikut: Laju Sampling = 40 kHz, frekuensi passband = 4 kHz, frekuensi stopband = 8 kHz, passband ripple = 0.5 dB, dan redaman stopband minimum = 40 dB.
   Catatan: Normalisasi nilai frekuensi dalam radian, sperti Wp =(4/40) Hz, Ws=(8/40) Hz
   Berikan penjelasan terhadap hasil yang diperoleh**.**
2. Ketikan command Matlab diatas untuk menghitung order terendah **filter highpass** IIR pada ke-4 jenis filter, menggunakan spesifikasi filter berikut: Laju Sampling = 3.500 Hz, frekuensi passband = 1.050 Hz, frekuensi stopband = 600 Hz, passband ripple = 1 dB, dan redaman stopband minimum = 50 dB.
   Berikan 4penjelasan terhadap hasil yang diperoleh**.**
3. Ketikan command Matlab diatas untuk menghitung order terendah **filter bandpass** IIR pada ke-4 jenis filter, menggunakan spesifikasi filter berikut: Laju Sampling = 7 kHz, frekuensi passband = 1.4 kHz dan 2.1 kHz, frekuensi stopband = 1.05 kHz dan 2.45 kHz, passband ripple = 0.4 dB, dan redaman stopband minimum = 50 dB.
   Berikan penjelasan terhadap hasil yang diperoleh
4. Ketikan command Matlab diatas untuk menghitung order terendah **filter bandstop** IIR pada ke-4 jenis filter, menggunakan spesifikasi filter berikut: Laju Sampling = 12 kHz, frekuensi

passband = 2.1 kHz dan 4.5 kHz, frekuensi stopband = 2.7 kHz dan 3.9 kHz, passband ripple = 0.6 dB, dan redaman stopband minimum = 45 dB.

Berikan penjelasan terhadap hasil yang diperoleh

### 4.4.2 Implementasi Filter Low Pass IIR

Setelah tipe filter telah dipilih dan ordenya telah diestimasi, langkah berikutnya adalah menentukan fungsi transfer filter. Untuk mendisain filter digital Butterworth pada orde N, command matlabnya adalah:

[num,den] = butter(N,Wn,'high') filter Highpass

[num,den] = butter(N,Wn,'stop') filter bandstop

**Langkah Percobaan:**

1. Buat script Matlab dan simpan hasilnya dengan nama "IIR_LPF".

```
clc;
close all;
clear all;
format long
rp=input('enter the passband ripple :');
rs=input('enter stopband ripple :');
wp=input('enter passband freq :');
ws=input('enter stopband freq :');
fs=input('enter sampling freq :');
w1=2*wp/fs;
w2=2*ws/fs;

%Digital LPF
[n,wn]= buttord(w1,w2,rp,rs);
[b,a]=butter(n,wn);
w=0:.01:pi;
[h,om]=freqz(b,a,w);
m=20*log10(abs(h));
an=angle(h);
figure(1)
plot(om/pi,m);
title('**** Digital Output Magnitude *****');
ylabel('gain in db...>');
xlabel('normalised freq..>');
figure(4)
plot(om/pi,an);
title('**** Digital Output Phase ****');
xlabel('normalised freq..>');
ylabel('phase in radians...>');
```

2. Jelaskan prosedur dari sript Matlab diatas, disesuaikan dengan teori implementasi LPF IIR
3. Inputkan spesifikasi filter IIR dengan rincian sebagai berikut : Laju Sampling = 40 kHz, frekuensi passband = 4 kHz, frekuensi stopband = 8 kHz, passband ripple = 0.5 dB, dan redaman stopband minimum = 40 dB.

Berikan penjelasan terhadap hasil yang diperoleh.

4. Inputkan spesifikasi filter IIR dengan rincian sebagai berikut : Laju Sampling = 10.000 Hz, frekuensi passband = 1500 Hz, frekuensi stopband = 3000 Hz, passband ripple = 0.5 dB, dan redaman stopband minimum = 100 dB.

   Bandingkan hasil yang diperoleh dengan percobaan 3.

### 4.4.3 Implementasi Filter High Pass IIR

Langkah Percobaan:

1. Buat script Matlab dan simpan hasilnya dengan nama "IIR_HPF".

```
clc;
close all;
clear all;
format long
rp=input('enter the passband ripple :');
rs=input('enter stopband ripple :');
wp=input('enter passband freq :');
ws=input('enter stopband freq :');
fs=input('enter sampling freq :');
w1=2*wp/fs;
w2=2*ws/fs;

%Digital HPF
[n,wn]= buttord(w1,w2,rp,rs);
[b,a]=butter(n,wn,'high');
w=0:.01:pi;
[h,om]=freqz(b,a,w); m=20*log10(abs(h));
an=angle(h);
figure(3)
plot(om/pi,m);
title('**** Digital Output Magnitude *****');
ylabel('gain in db...>');
xlabel('normalised freq..>');
figure(4)
plot(om/pi,an);
title('**** Digital Output Phase ****');
xlabel('normalised freq..>');
ylabel('phase in radians...>');
```

2. Inputkan spesifikasi filter IIR dengan rincian sebagai berikut: Laju Sampling = 3.500 Hz, frekuensi passband = 1.050 Hz, frekuensi stopband = 600 Hz, passband ripple = 1 dB, dan redaman stopband minimum = 50 dB.

   Berikan penjelasan terhadap hasil yang diperoleh.

3. Inputkan spesifikasi filter IIR dengan rincian sebagai berikut: Laju Sampling = 8.000 Hz, frekuensi passband = 1.200 Hz, frekuensi stopband = 2400 Hz, passband ripple = 0.5 dB, dan redaman stopband minimum = 100 dB.

Bandingkan hasil yang diperoleh dengan percobaan 3.

## 4.5 IMPLEMENTASI LPF FIR

Langkah Percobaan:

1. Buat script Matlab dan simpan hasilnya dengan nama “FIR_LPF”.

```
clc;
close all;
clear all;
rp=input('enter the passband ripple :');
rs=input('enter the stopband ripple :') ;
fp=input('enter the passband frequency :');
fs=input('enter the stopband frequency :');
f=input('enter the sampling freq :');
wp=2*fp/f;
ws=2*fs/f;
num=-20*log10(sqrt(rp*rs))-13;
dem=14.6*(fs-fp)/f;
n=ceil(num/dem);
n1=n+1;
if(rem(n,2)~=0)
      n1=n;
      n=n-1;
end y=boxcar(n1);
b=fir1(n,wp,y);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
an=angle(h);
figure(1)
plot(o/pi,m);
title('******** LOW PASS FIR FILTER RESPONSE
********');
ylabel('GAIN in db--->');
xlabel('Normalised Frequency--->');
figure(2)
plot(o/pi,an);
title('******** LOW PASS FIR FILTER RESPONSE
********');
ylabel('PHASE--->');
xlabel('Normalised Frequency--->');
```

2. Jelaskan prosedur dari sript Matlab diatas, disesuaikan dengan teori implementasi LPF FIR
3. Inputkan spesifikasi filter IIR dengan rincian sebagai berikut: Laju Sampling = 8.000 Hz, frekuensi passband = 1.500 Hz, frekuensi stopband = 2.000 Hz, passband ripple = 0.05 dB, dan redaman stopband minimum = 0.04 dB.

   Berikan penjelasan terhadap hasil yang diperoleh, dan cobakan dengan spesifikasi yang lain.

**Tugas :**

1. Buatlah script matlab untuk implementasi HPF FIR dan ujikan dengan beberapa spesifikasi inputan filter.
2. Tambahkan tinjaun teori untuk disain filter IIR dan FIR