



MODUL III

IMPLEMENTASI FILTER IIR DAN FIR

TUJUAN PRAKTIKUM :

1. Memahami perbedaan antara filter IIR dan FIR
2. Merancang filter IIR dan FIR melalui perhitungan
3. Mengimplementasikan filter IIR dan FIR dalam bentuk *equalizer 3 band* secara sederhana dengan menggunakan MATLAB

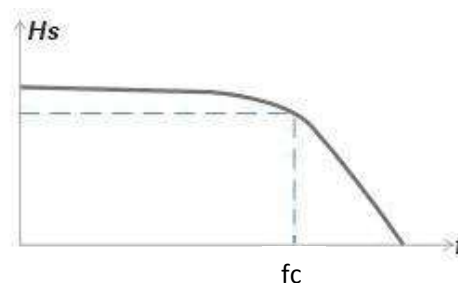
DASAR TEORI

1. FILTER

Dalam pemrosesan sinyal digital, filter digunakan untuk memisahkan bagian-bagian yang tidak diinginkan dari suatu sinyal, seperti *noise*, dengan cara meredam atau melewatkan frekuensi tertentu, sehingga diperoleh sinyal yang diinginkan. Filter merupakan suatu sistem yang mempunyai fungsi transfer tertentu untuk meloloskan sinyal masukan pada frekuensi-frekuensi tertentu dan menyaring / memblokir / melemahkan sinyal masukan pada frekuensi-frekuensi yang lain.

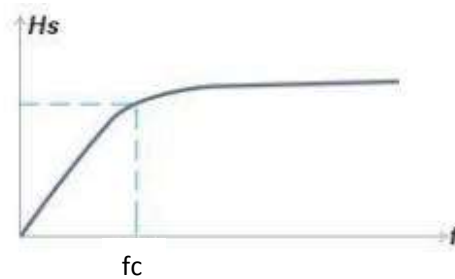
Filter diklasifikasikan :

- a. Berdasarkan frekuensi yang disaring :
 - *Low Pass Filter* : Filter yang melewatkan frekuensi yang lebih rendah dari frekuensi *cut-off* (f_c).



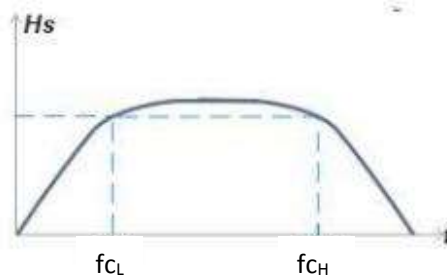
Gambar 1.LPF

- *High Pass Filter* : Filter yang melewatkan frekuensi yang lebih tinggi dari frekuensi *cut-off* (f_c). Biasanya frekuensi *cut-off* berada di nilai -3 dB.



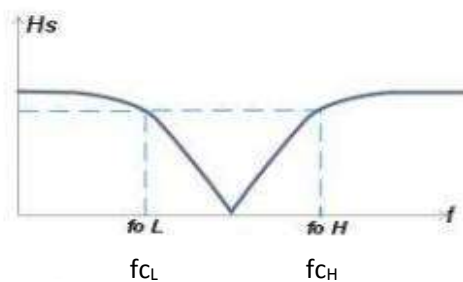
Gambar 2.HPF

- *Band Pass Filter* : Filter yang melewatkan suatu *range* frekuensi diantara dua frekuensi *cut off* (f_c), yaitu f_{cL} dan f_{cH} .



Gambar 3.BPF

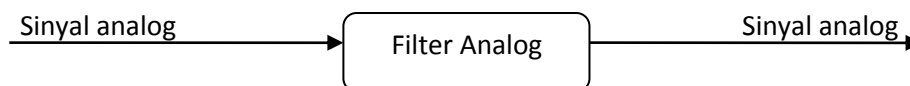
- *Band Stop Filter* : Filter yang melewatkan frekuensi diluar *range* frekuensi yang berada diantara dua f_c yaitu f_{cL} dan f_{cH} .



Gambar 4.BSF

b. Berdasarkan inputannya, terdapat dua jenis filter, yaitu filter analog dan filter digital.

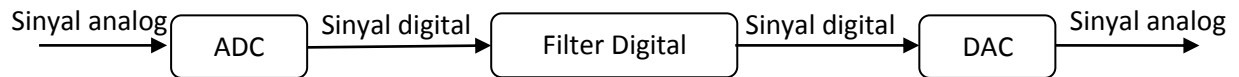
- Filter analog : sinyal masukan berupa sinyal analog



Gambar 5. Blok Filter Analog



- Filter digital : sinyal masukan berupa sinyal diskrit



Gambar 6. Blok Filter Digital

Filter digital terdiri atas:

- ❖ FIR (*Finite Impulse Response*)
- ❖ IIR (*Infinite Impulse Response*)

2. FIR (*FINITE IMPULS RESPONSE*)

Perbedaan yang mendasar antara Filter IIR dan FIR adalah pemakaian filter FIR terbatas untuk waktu diskrit. Perancangan filter FIR berdasarkan aproksimasi langsung respon frekuensi yang diinginkan dari sistem waktu diskrit. FIR merupakan suatu sistem yang non-rekursif, dimana keluaran saat ini tidak bergantung pada keluaran sebelumnya, sehingga mempunyai respon impuls yang terbatas. FIR dapat dirancang agar kausal dan mempunyai fase linier. Filter fase linier yaitu filter yang mempunyai respon impuls yang simetri di kiri dan kanan sumbu y .

Ada dua metode untuk mendesain filter FIR, yaitu :

- a. Metode frekuensi sampling
- b. Metode *windowing*

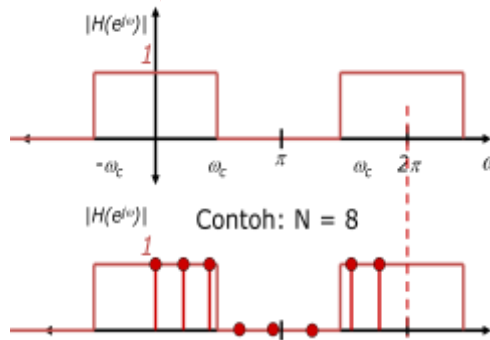
2.1 Metode Fekuensi Sampling

Metode sampling frekuensi dilakukan dengan mensampel N titik dari respons frekuensi yang diinginkan pada interval $0-2\pi$. Koefisien yang diperoleh ($H(k)$) kemudian diinvers transformasi Fourierkan untuk memperoleh $h(n)$. (IDFT ada di modul dua).

Langkah perancangannya:

1. Sketsa respon frekuensi filter $\rightarrow H(e^{j\omega}) \rightarrow 0 \leq \omega \leq 2\pi$
2. Sampling sejumlah N titik $\rightarrow H(k)$
3. IDFT [$H(k)$] $\rightarrow h(n)$

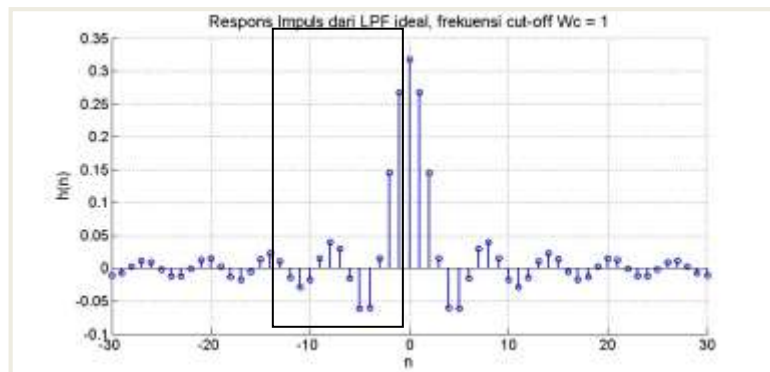
4. Realisasikan



Gambar 7. $|H(e^{j\omega})|$

2.2 Metode Windowing

Metode *windowing* merupakan metode yang sederhana untuk mendesain filter FIR, serta mudah untuk digunakan. Cara termudah untuk menghasilkan Filter FIR adalah dengan memotong respon impuls dari filter ideal kemudian menggesernya sehingga menjadi kausal. Proses ini dinamakan *windowing*.



Gambar 8. Respon Impuls dari LPF ideal

Efek windowing di domain frekuensi adalah konvolusi dari respon frekuensi ideal dengan respon frekuensi dari fungsi *window*. Adapun jenis-jenis *window* terdiri dari : *rectangular, bartlett, hamming, hanning, blackman, kaiser, lancroz, turkey*, dll.



Langkah-langkah Perancangan Filter FIR dengan Metode *Windowing*

1. Tentukan Respon Impuls Filter Ideal

$$\begin{aligned} \text{LPF } h(n) &= \frac{1}{2\pi} \int_{-\omega_{c1}}^{\omega_{c1}} e^{j\omega n} d\omega \\ &= \frac{1}{\pi n} \sin(\omega_c n) \\ \text{HPF } h(n) &= \frac{1}{2\pi} \left(\int_{-\pi}^{-\omega_{c1}} e^{j\omega n} d\omega + \int_{\omega_{c1}}^{\pi} e^{j\omega n} d\omega \right) \\ &= \frac{1}{\pi n} \sin(\pi n) - \frac{1}{\pi n} \sin(\omega_c n) \\ \text{BPF } h(n) &= \frac{1}{2\pi} \left(\int_{-\omega_{c2}}^{-\omega_{c1}} e^{j\omega n} d\omega + \int_{\omega_{c1}}^{\omega_{c2}} e^{j\omega n} d\omega \right) \\ &= \frac{1}{\pi n} \sin(\omega_{c2} n) - \frac{1}{\pi n} \sin(\omega_{c1} n) \\ \text{BSF } h(n) &= \frac{1}{\pi} \left(\int_0^{\omega_{c1}} e^{j\omega n} d\omega + \int_{\omega_{c2}}^{\pi} e^{j\omega n} d\omega \right) \\ &= \frac{1}{\pi n} \sin(\omega_{c1} n) + \frac{1}{\pi n} \sin(\pi n) - \frac{1}{\pi n} \sin(\omega_{c2} n) \end{aligned}$$

(2.1)

2. Tentukan window yang akan digunakan, berdasarkan redaman *stopband* filter yang diinginkan.

No	Window	LebarPitaTransisi	Maksimum Redaman <i>Stopband</i> (dB)
1	<i>Rectangular</i>	$4\pi/N$	-21
2	<i>Bartlett</i>	$8\pi/N$	-25
3	<i>Hanning</i>	$8\pi/N$	-44
4	<i>Hamming</i>	$8\pi/N$	-53
5	<i>Blackman</i>	$12\pi/N$	-74

Tabel 1. Jenis *window*

3. Berdasarkan lebar pita transisi, tentukan Lebar window N (orde minimum)

Rectangular	: $\omega_R(n) = 1, \quad 0 \leq n \leq N-1$
Bartlett	: $w_B(n) = \begin{cases} \frac{2n}{N-1} & \text{untuk } 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1} & \text{untuk } \frac{N-1}{2} \leq n \leq N-1 \\ 0 & \text{untuk yang lain} \end{cases}$
Hanning	: $\omega_{Han}(n) = 0.5 - 0.5\cos[2\pi n/(N-1)], \quad 0 \leq n \leq N-1$
Hamming	: $\omega_{Ham}(n) = 0.54 - 0.46\cos[2\pi n/(N-1)], \quad 0 \leq n \leq N-1$
Blackman	: $\omega_{Bl}(n) = 0.42 - 0.5\cos[2\pi n/(N-1)] + 0.08\cos[4\pi n/(N-1)], \quad 0 \leq n \leq N-1$

(2.2)

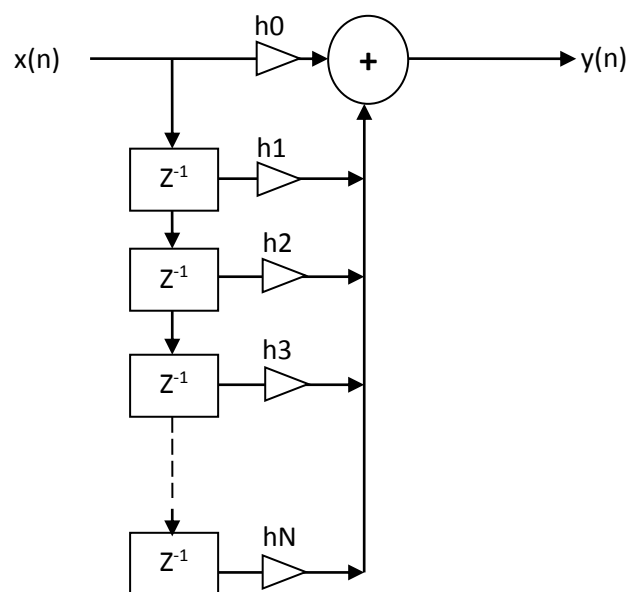
4. Tentukan Respon Impuls Sistem

$$h(n) = h(n) * \omega(n)$$

(2.3)

Respon Impuls Filter : $h(n) = [h_0, h_1, h_2, h_3, \dots, h_N]$

5. Realisasi Sistem



Gambar 9. Realisasi Sistem



Contoh perancangan filter FIR metode windowing

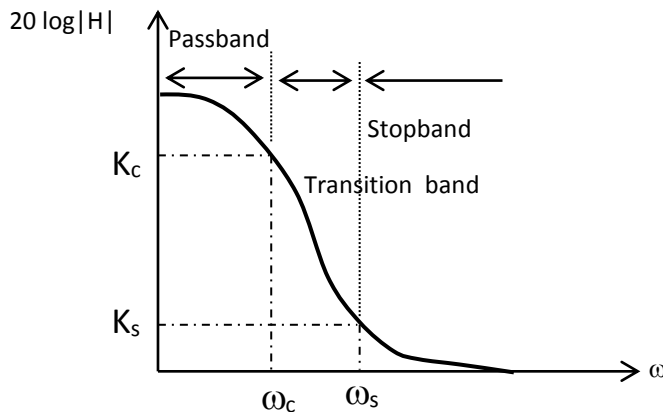
Rencanakan filter digital *low-pass non-recursive* yang mempunyai karakteristik sebagai berikut : Amplitudo pada pita frekuensi lolos turun sampai -3 dB dari 0 s/d 1000 Hz. Pada frekuensi 3300 Hz diinginkan sinyal diredam minimum 24 dB. Filter tersebut direncanakan bekerja pada frekuensi pencuplikan 10 KHz.

- a) Tentukan persamaan response impuls $h(n)$
- b) Gambarkan realisasi rangkaian filter digital tersebut.
- c) Tentukan persamaan fungsi transfer $H(z)$ dan persamaan selisihnya dari hasil perencanaan.
- d) Tentukan persamaan *response* frekuensi $H(e^{j\omega})$ filter digital hasil perencanaan.

Penyelesaian:

i. Sketsa filter rancangannya

Sketsa *Response magnitude* filter *LPF* dapat dilihat seperti pada gambar berikut:



Gambar 10. Respon frekuensi LPF

Dari gambar di atas terlihat bahwa ada tiga daerah, yaitu daerah *passband*, daerah *transitionband*, dan daerah *stopband*. Prosedur untuk mencari nilai parameter (ω_c, ω_s , orde filter) adalah sebagai berikut:

$$\omega_c = \frac{2\pi \times f_c}{f_{sampling}} = \frac{2\pi \times 1000}{10.000} = 0,2\pi \text{ rad} \quad \text{dan} \quad \omega_s = \frac{2\pi \times f_s}{f_{sampling}} = \frac{2\pi \times 3300}{10.000} = 0,66\pi \text{ rad}$$



ii. Menentukan respon impuls ideal atau $h_{id}(n)$ filter

Karena akan merancang filter LPF, maka dari tabel $h_{id}(n)$ didapat respon impuls idealnya:

$$h_{id}(n) = \frac{\sin(\omega_c n)}{\pi n} = \frac{\sin(0,2\pi n)}{\pi n}$$

iii. Penentuan *window dan orde filter*

Pemilihan jenis *window* yang digunakan dapat dilakukan dengan memperhatikan tabel. Dari tabel di atas untuk mendapatkan redaman sebesar -24 dB atau lebih pada daerah *stopband*, maka dapat digunakan *window Bartlett, Hanning, Hamming*, atau *Blackman*.

Orde filter(N) dapat ditentukan dengan menggunakan tabel *window* diatas dengan memanfaatkan lebar transisinya. Sehingga dapat diperoleh transisi *band* adalah:

$$\omega_c = 0,2 \pi \text{ dan } \omega_s = 0,66 \pi$$

$$\text{Transisi } band = |\omega_c - \omega_L| = |0,2\pi - 0,66\pi| = 0,46\pi$$

Dari lebar transisi masing-masing *window* yang memenuhi, dapat diperoleh orde filter N perancangan sebagai berikut:

Window	Lebar Transisi	Pelemahan <i>stoband</i> minimum	Orde filter (N)
<i>Bartlett</i>	$8 \pi / N$	-25 dB	$\frac{8\pi}{N} = 0,46\pi \Rightarrow N = \frac{8\pi}{0,46\pi} = 18$
<i>Hanning</i>	$8 \pi / N$	-44 dB	$\frac{8\pi}{N} = 0,46\pi \Rightarrow N = \frac{8\pi}{0,46\pi} = 18$
<i>Hamming</i>	$8 \pi / N$	-53 dB	$\frac{8\pi}{N} = 0,46\pi \Rightarrow N = \frac{8\pi}{0,46\pi} = 18$
<i>Blackman</i>	$12 \pi / N$	-74 dB	$\frac{12\pi}{N} = 0,46\pi \Rightarrow N = \frac{12\pi}{0,46\pi} = 26$

Spesifikasi perancangan filter pada *stopband* pada -24dB, maka dari spesifikasi *windowing* tabel tersebut, *window* Barlett yang paling mendekati dan orde filter N yang minimum. Selanjutnya, untuk mempermudah perancangan dan perhitungan, kita gunakan orde filter yang bernilai ganjil. Sehingga $N=18+1=19$.



iv. Impuls response h(n)

Pergeseran α dengan orde filter N dapat ditentukan dengan persamaan $\frac{N-1}{2}$ sehingga $\alpha = \frac{19-1}{2} = 9$

Impuls respons $h(n)$:

$$h(n) = h_{id}(n - \alpha) \cdot w_B(n) = \frac{\sin[\omega_c(n - \alpha)]}{\pi[n - \alpha]} \cdot w_B(n) = \frac{\sin[\omega_c(n - 9)]}{\pi[n - 9]} \cdot w_B(n)$$

dimana w_B adalah *window Bartlett* yang memiliki persamaan sebagai berikut:

$$w_B(n) = \begin{cases} \frac{2n}{N-1} & \text{untuk } 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1} & \text{untuk } \frac{N-1}{2} \leq n \leq N-1 \\ 0 & \text{untuk yang lain} \end{cases}$$

Didapat nilai $w_B(n) = \begin{cases} \frac{n}{9} & \text{untuk } 0 \leq n \leq 9 \\ 2 - \frac{n}{9} & \text{untuk } 9 \leq n \leq 18 \\ 0 & \text{untuk yang lain} \end{cases}$

Sehingga,

$$h(n) = \begin{cases} \frac{n}{9} \cdot \frac{\sin[0,2\pi(n-9)]}{\pi[n-9]} & \text{untuk } 0 \leq n \leq 9 \\ (2 - \frac{n}{9}) \cdot \frac{\sin[0,2\pi(n-9)]}{\pi[n-9]} & \text{untuk } 9 < n \leq 18 \end{cases}$$

$$h(n) = \begin{cases} [0 \ 0,0008 \ 0,0013 \ \dots] & \text{untuk } 0 \leq n \leq 9 \\ [\dots \ -0,0022 \ -0,0035 \ 0] & \text{untuk } 9 < n \leq 18 \end{cases}$$

Persamaan beda :

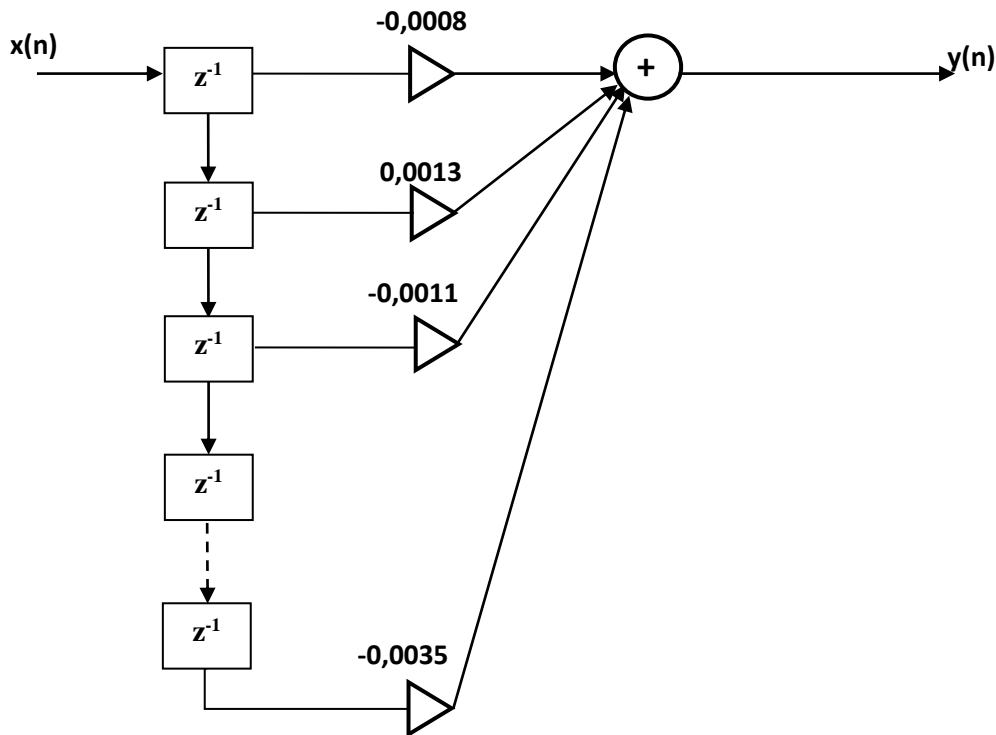
$$y(n) = \sum_{k=0}^{18} h(k) \cdot x(n-k)$$

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \dots + h(18)x(n-18)$$

Maka,

$$y(n) = 0 - 0,0008 x(n-1) + 0,0013 x(n-2) - 0,0011 x(n-3) + \dots - 0,0022 x(n-16) - 0,0035 x(n-17) + 0$$

v. Gambar realisasi rangkaian filter



Gambar 11. Realisasi rangkaian filter

3. IIR (Infinite Impuls Response)

Filter digital IIR memiliki respon impuls durasi tak terbatas, sehingga dapat ditransformasikan dari filter analog yang umumnya memiliki respon impuls yang panjangnya tidak terbatas. Respon impuls yang tidak terbatas dapat dipahami sebagai suatu filter yang memperhitungkan kondisi sebelum dan sesudahnya sebagai gabungan antara *feedback* dan *feedforward*.

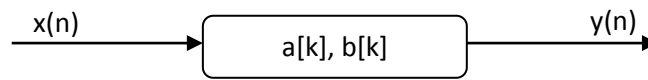
Transformasi tersebut dapat dilakukan setidaknya dengan tiga cara, yaitu :

1. Transformasi Z, yaitu dengan cara mendekati persamaan beda filter analog dengan persamaan beda filter digital.

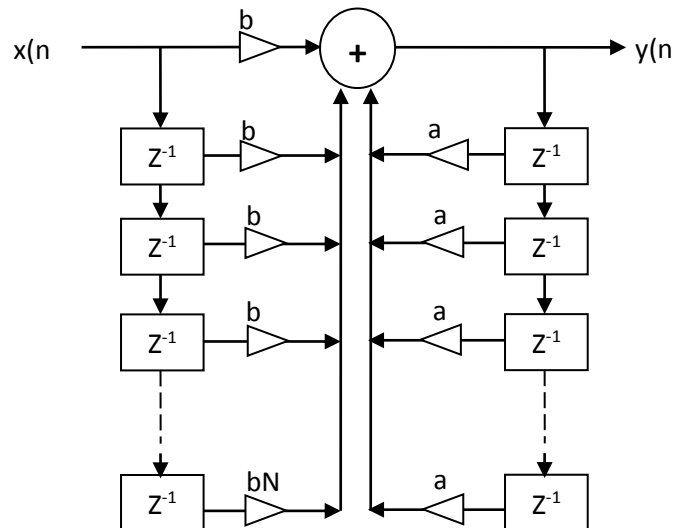
2. Impuls invariant, yaitu dengan cara mencuplik respon impuls dari sebuah sistem analog dengan jarak cuplik yang sama. Metode transformasi z dan impuls invariant memiliki keterbatasan yaitu hanya efektif untuk perancangan filter *lowpass* dan *bandpass* saja.
3. Transformasi bilinear, yaitu metode yang dapat mengatasi keterbatasan pada metode transformasi z dan impuls invariant.

Karakteristik dari IIR adalah sebagai berikut :

1. Respon impuls bernilai tak hingga
2. Belum tentu stabil
 - *Pole* di dalam lingkaran satuan → stabil
 - *Pole* di luar lingkaran satuan → tidak stabil
3. Harus mengandung *pole*
4. Tidak harus mengandung *zero*.



Gambar12. Filter IIR



Gambar 13. Realisasi filter IIR

Persamaan selisih filter IIR

$$y[n] = \sum_{k=0}^N b[k]x[n-k] + \sum_{k=1}^M a[k]y[n-k]$$



(3.1)

Ket :

b= koefisien *feedback*

Banyaknya (total koefisien) = M+N+1

a = koefisien *feedforward*

N = orde filter IIR

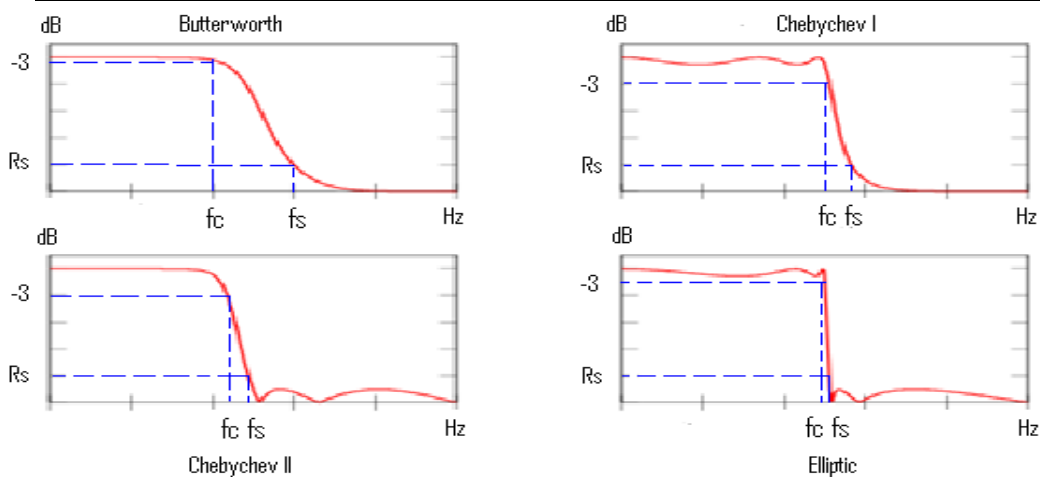
3.1 Pendekatan Filter Analog :

- Filter *Butterworth*
- Filter *Chebyshev*
- Filter *Elliptic*

Pendekatan filter analog dapat dilihat pada tabel 2.

Tabel 2. Pendekatan Filter Analog

Pendekatan	<i>Passband</i>	<i>Stopband</i>
<i>Butterworth</i>	<i>No Ripple</i>	<i>No Ripple</i>
<i>Chebyshev I</i>	<i>Ripples</i>	<i>No Ripple</i>
<i>Chebyshev II</i>	<i>No Ripple</i>	<i>Ripples</i>
<i>Elliptic</i>	<i>Ripples</i>	<i>Ripples</i>



Gambar 14. Respon frekuensi dari Pendekatan Filter Analog

Respon Prototipe	Respon Filter hasil transformasi pita frekuensi	Persamaan Perancangan
<p>Low-pass $G(s)$</p>	<p>Low-pass $H(s)$</p>	<p>Arah Maju/Forward : $\Omega_s' = \Omega_s \Omega_p$</p> <p>Arah Mundur/Backward :</p> $\Omega_s = \Omega_s' / \Omega_p$
<p>Low-pass $G(s)$</p>	<p>High-pass $H(s)$</p>	<p>Arah Maju/Forward : $\Omega_s' = \Omega_p / \Omega_s$</p> <p>Arah Mundur/Backward :</p> $\Omega_s = \Omega_p / \Omega_s'$
<p>Low-pass $G(s)$</p>	<p>Band-pass $H(s)$</p>	<p>Arah Maju/Forward : $\Omega_R = \frac{\Omega_{p2} - \Omega_{p1}}{2}$</p> $\Omega_1 = \sqrt{\Omega_s^2 \Omega_R^2 + \Omega_{p1} \Omega_{p2}} - \Omega_R \Omega_s$ $\Omega_2 = \sqrt{\Omega_s^2 \Omega_R^2 + \Omega_{p1} \Omega_{p2}} + \Omega_R \Omega_s$ <p>Arah Mundur/Backward :</p> $\Omega_s = \min\{ A , B \}$ $A = \frac{-\Omega_{s1}^2 + \Omega_{p1} \Omega_{p2}}{\Omega_{s1}(\Omega_{p2} - \Omega_{p1})}$ $B = \frac{-\Omega_{s2}^2 + \Omega_{p1} \Omega_{p2}}{\Omega_{s2}(\Omega_{p2} - \Omega_{p1})}$
<p>Low-pass $G(s)$</p>	<p>Band-stop $H(s)$</p>	<p>Arah Maju/Forward : $\Omega_R = \frac{\Omega_{p2} - \Omega_{p1}}{2}$</p> $\Omega_1 = \sqrt{(\Omega_R / \Omega_s)^2 + \Omega_{p1} \Omega_{p2}} - \Omega_R / \Omega_s$ $\Omega_2 = \sqrt{(\Omega_R / \Omega_s)^2 + \Omega_{p1} \Omega_{p2}} + \Omega_R / \Omega_s$ <p>Arah Mundur/Backward :</p> $\Omega_s = \min\{ A , B \}$ $A = \frac{\Omega_{s1}(\Omega_{p2} - \Omega_{p1})}{-\Omega_{s1}^2 + \Omega_{p1} \Omega_{p2}}$ $B = \frac{\Omega_{s2}(\Omega_{p2} - \Omega_{p1})}{-\Omega_{s2}^2 + \Omega_{p1} \Omega_{p2}}$

TABEL 3. Transformasi Butterworth Analog

Langkah-langkah Perancangan Filter IIR dengan Transformasi Bilinear

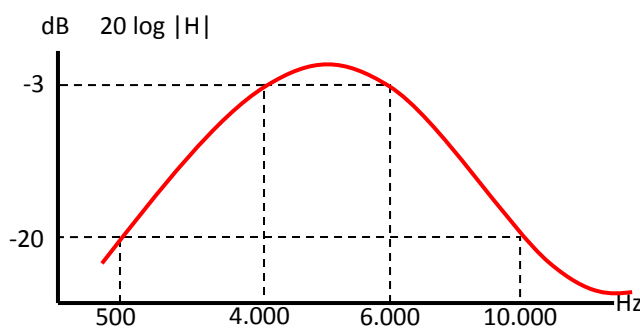
Desain filter IIR BPF dengan spesifikasi:

- Amplitudo rata pada frekuensi *passband*, di luar *passband* redaman monoton turun
- Redaman-3dB pd frekuensi 4000 Hz dan 6000 Hz
- Redaman -20 dB sinyal pada frekuensi 500 Hz dan 10.000 Hz
- Frekuensi Sampel = 22.000 Hz
- Menggunakan transformasi bilinear

Rancanglah filter digital sesuai dengan spesifikasi diatas !

Langkah Perancangan :

i. Gambar respon filter yang diinginkan



Jenis Filter: *Band Pass Filter*

Tipe Filter : *BPF Butterworth*

Gambar 15. Respon frekuensi LPF

ii. Menentukan frekuensi digital dengan menggunakan persamaan $\omega = \frac{2\pi.f}{F_s}$

$$\omega_{PL} = \frac{2\pi \times f_{PL}}{F_s} = \frac{2\pi \times 4.000}{22.000} = 0,3636\pi \quad \omega_{SL} = \frac{2\pi \times f_{SL}}{F_s} = \frac{2\pi \times 500}{22.000} = 0,0455\pi$$

$$\omega_{PH} = \frac{2\pi \times f_{PH}}{F_s} = \frac{2\pi \times 6.000}{22.000} = 0,5455\pi \quad \omega_{SH} = \frac{2\pi \times f_{SH}}{F_s} = \frac{2\pi \times 10.000}{22.000} = 0,9091\pi$$

iii. Konversi frekuensi digital ke frekuensi analog (prewarping)

Konversi menggunakan mode radian dengan persamaan $\Omega = 2.Fs.\tan\frac{\omega}{2}$

$$\Omega_{PL} = 2Fs.\tan\frac{\omega_{PL}}{2} = 2Fs.0,6427 \quad \Omega_{SL} = 2Fs.\tan\frac{\omega_{SL}}{2} = 2Fs.0,0715$$

$$\Omega_{PH} = 2Fs.\tan\frac{\omega_{PH}}{2} = 2Fs.1,1541 \quad \Omega_{SH} = 2Fs.\tan\frac{\omega_{SH}}{2} = 2Fs.6,9552$$



iv. Menentukan filter prototype LPF untuk BPF Butterworth

$$A = \frac{-\Omega_{SL}^2 + \Omega_{PL}\Omega_{PH}}{\Omega_{SL}(\Omega_{PH} - \Omega_{PL})} = \frac{-(2Fs.0,0715)^2 + 2Fs.0,6427 \times 2Fs.0,1,1541}{2Fs.0,0715 \times (2Fs.0,6427 - 2Fs.0,1,1541)} = 20,1376$$

$$B = \frac{-\Omega_{SH}^2 + \Omega_{PL}\Omega_{PH}}{\Omega_{SH}(\Omega_{PH} - \Omega_{PL})} = \frac{-(2Fs.6,9552)^2 + 2Fs.0,6427 \times 2Fs.0,1,1541}{2Fs.6,9552 \times (2Fs.0,6427 - 2Fs.0,1,1541)} = -13,3917$$

$$\Omega_c = \min\{|A|, |B|\} = \min\{|20,1376|, |-13,3917|\} = 13,3917$$

v. Menentukan orde filter (N)

$$n = \left\lceil \frac{\log\left(\frac{10^{-Rp/10} - 1}{10^{-Rs/10} - 1}\right)}{2 \log\left(\frac{1}{\Omega_c}\right)} \right\rceil = \left\lceil \frac{\log\left(\frac{10^{0,3} - 1}{10^2 - 1}\right)}{2 \log\left(\frac{1}{13,3917}\right)} \right\rceil = \lceil 0,8864 \rceil = 1$$

vi. Denormalisasi

Sesuai table, untuk orde 1 pada Butterworth, maka $H_n(S) = \frac{1}{s+1}$

Untuk BPF Butterwoth, maka

$$H(S) = H_n(S) \Big|_{S = \frac{S^2 + \Omega_{PL}\Omega_{PH}}{S(\Omega_{PH} - \Omega_{PL})} = \frac{S^2 + 0,7417}{0,5114.S}} = \frac{1}{\frac{S^2 + 0,7417}{0,5114.S} + 1} = \frac{0,5114.S}{S^2 + 0,5114.S + 0,7417}$$

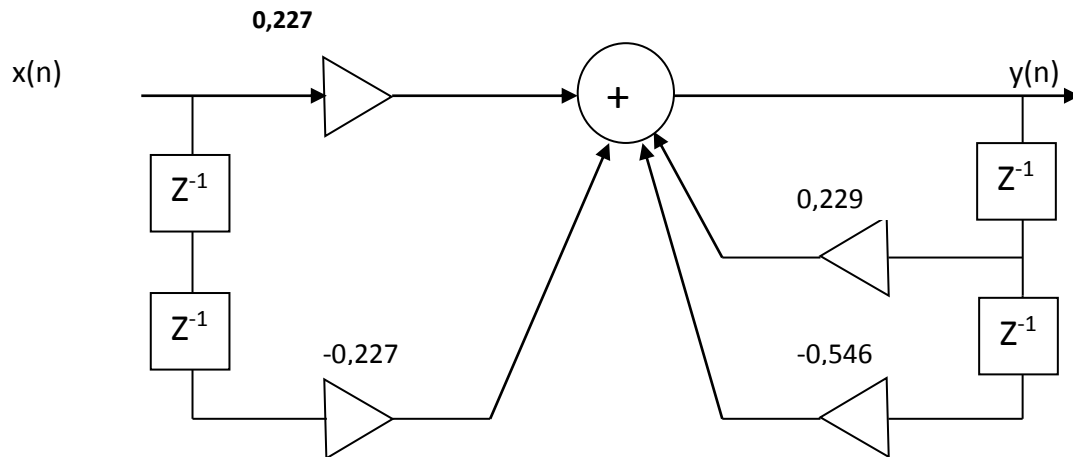
vii. Menentukan H(z) dengan transformasi bilinear

$$H(z) = H(s) \Big|_{s = \frac{z-1}{z+1}}$$

$$\begin{aligned} H(z) &= \frac{0,5114.S}{S^2 + 0,5114.S + 0,7417} \Big|_{s = \frac{z-1}{z+1}} \\ &= \frac{0,5114 \cdot \left(\frac{z-1}{z+1}\right)}{\left(\frac{z-1}{z+1}\right)^2 + 0,5114 \cdot \left(\frac{z-1}{z+1}\right) + 0,7417} \\ &= \frac{0,227 - 0,227.Z^{-2}}{1 - 0,229.Z^{-1} + 0,546.Z^{-2}} \end{aligned}$$

viii. Gambar Realisasi sistem

$$Y(Z) = 0,227 \cdot X(Z) - 0,227 \cdot Z^{-2} X(Z) + 0,229 \cdot Z^{-1} Y(Z) - 0,546 \cdot Z^{-2} Y(Z)$$



Gambar 16. Realisasi filter dengan orde 1

Keuntungan IIR dibandingkan dengan FIR :

- Pada filter IIR orde tinggi dapat menghasilkan respon frekuensi yang tajam / curam (hampir mendekati ideal).
- Membutuhkan koefisien yang lebih sedikit untuk respon frekuensi yang curam sehingga dapat mengurangi jumlah waktu komputasi.

Keuntungan FIR dibandingkan IIR:

- Selalu stabil
- Dapat dirancang untuk memiliki fase yang linier
- Lebih mudah diimplementasikan ke *hardware*
- Respon impuls memiliki durasi yang terbatas.



4. EQUALIZER

Equalizer adalah rangkaian yang mampu mengamplifikasi (menguatkan) atau mengatenuasi (meredam) rentang frekuensi tertentu dan membiarkan yang lain tetap utuh.

4.1 Jenis equalizer:

1. ***low cut filter/hi pass***, jenis ini berfungsi untuk memotong sinyal frekuensi rendah/ meloloskan frekuensi tinggi sesuai batas frekuensi yang ditentukan.
2. ***hi cut filter/low pass***, jenis ini kebalikan dari *low cut/hi pass* dimana perbedaanya frekuensi tinggi di-*cut* sesuai batas yang ditentukan.
3. ***band pass filter (BPF)***, memungkinkan sinyal masuk untuk diproses dB-nya (*boost/cut*) sesuai dengan range frekuensi yang diinginkan.
4. ***shelving equalizer/oneband equalizer***, jenis ini hanya terdapat satu parameter yang berfungsi untuk mengontrol dB (*boost/cut*) dari frekuensi yang di tentukan. Jadi hanya frekuensi yang tertentu saja yang bisa diproses sesuai yang sudah tertulis pada parameter.
5. ***sweeping equalizer***, pada umumnya sama seperti *shelving equalizer* yaitu menambah atau mengurangi kekerasan suara dari ferkuensi. Perbedaanya, pada *sweeping equalizer* ini mempunyai 2 parameter, parameter yang pertama mempunyai fungsi menentukan frekuensi berapa yang akan diproses dan yang kedua menentukan berapa dB yang akan ditambah maupun dikurang. Jadi frekuensi yang akan diproses dapat diubah-ubah / ditentukan sesuai keinginan. Jenis equalizer ini biasanya ada pada *mixer-mixer* profesional.
6. ***parametric equalizer***, adalah jenis *equalizer* yang mempunyai 3 parameter, yaitu parameter untuk mengatur kekerasan suara, parameter untuk mengatur frekuensi(Hz), selanjutnya parameter yang mengatur kelebaran frekuensi/bandwith yang akan ikut terpengaruh.

4.2 Fungsi equalizer:

1. Meminimalkan atau membuang karakter yang tidak diinginkan.
2. Menonjolkan karakter tertentu dari suatu instrumen.



3. Menempatkan setiap instrumen pada lagu supaya terjadi keseimbangan / *balance frequency*

4.3 Perancangan *3-band equalizer* berdasarkan filter yang digunakan menggunakan Matlab

Pada Matlab, dalam merancang filter digital IIR maupun FIR telah tersedia beberapa *syntax* yang dapat digunakan

Filter IIR :

- *Butterworth* :
`>> [b,a] = butter(orde, Wn, 'tipe_filter')`
- *Elliptic* :
`>> [b,a] = ellip(orde, Ripplepass, Ripplestop, Wn, 'tipe_filter')`
- *Chebyshev 1* :
`>> [b,a] = cheby1(orde, Ripplepass, Wn, 'tipe_filter')`
- *Chebyshev 2* :
`>> [b,a] = cheby2(orde, Ripplestop, Wn, 'tipe_filter')`

Filter FIR

```
>> [b,a] = fir1(orde,Wn,'tipe_filter',window(orde+1))
```

Ket : (Wn = frekuensi *cut-off* ternormalisasi)

$Wn = (fc/(fs/2)) \rightarrow$ untuk *LPF* dan *HPF*

$Wn = ([fc1\ fc2]/(fs/2)) \rightarrow$ untuk *BPF* dan *BSF*

Tipe_filter :

- *high* \rightarrow *highpass filter*
- *low* \rightarrow *lowpass filter*
- *bandpass* \rightarrow *bandpass filter*
- *stop* \rightarrow *bandstop filter*

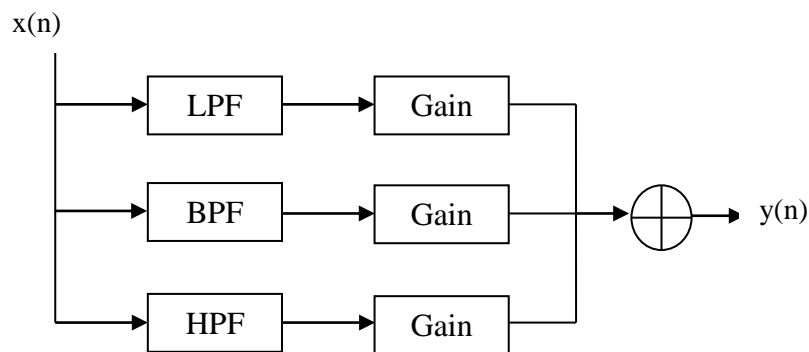


Window :

- o *rectwin* → *window rectangular*
- o *bartlett* → *window bartlett*
- o *hann* → *window hanning*
- o *hamming* → *window hamming*
- o *blackman* → *window blackman*

Cara kerja *equalizer* sendiri secara garis besarnya adalah membesarkan atau mengecilkan intensitas suara pada range frekuensi tertentu dengan besaran(*gain*) tertentu. Dalam *equalizer* pertama suara akan dikelompokkan dalam kelompok-kelompok frekuensi tertentu. Pengelompokan frekuensi sendiri dilakukan oleh filter digital. Suara yang sudah dikelompokkan dalam frekuensi tertentu tersebut kemudian akan diamplifikasi dengan besaran(*gain*) tertentu. Setelah diamplifikasi dengan *gain* tertentu, suara tersebut disatukan kembali untuk menghasilkan nada yang diinginkan.

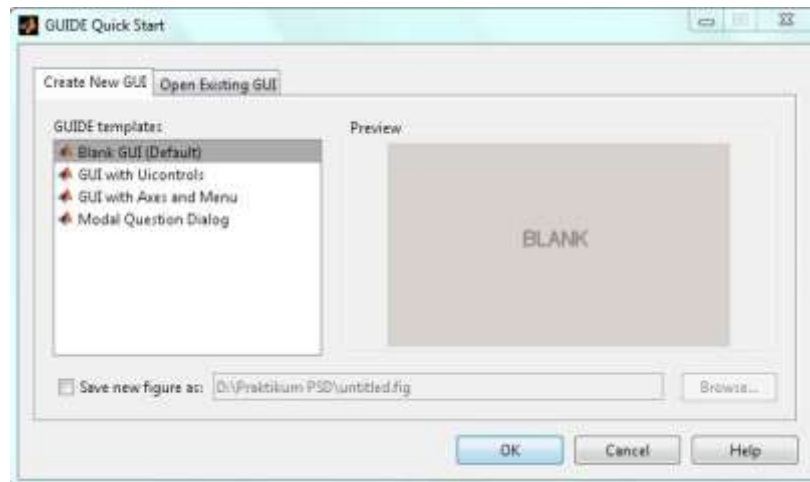
Pada modul ini, kita akan membuat *equalizer* dengan tiga buah *band*, artinya kita akan mempunyai tiga kelompok frekuensi (*band*). Pada modul ini juga akan diperkenalkan cara membuat GUI (*Graphical User Interface*) sederhana sehingga program yang dibuat nanti memiliki tampilan yang lebih baik.



Langkah-langkah membuat program *equalizer 3 band* dengan GUI:

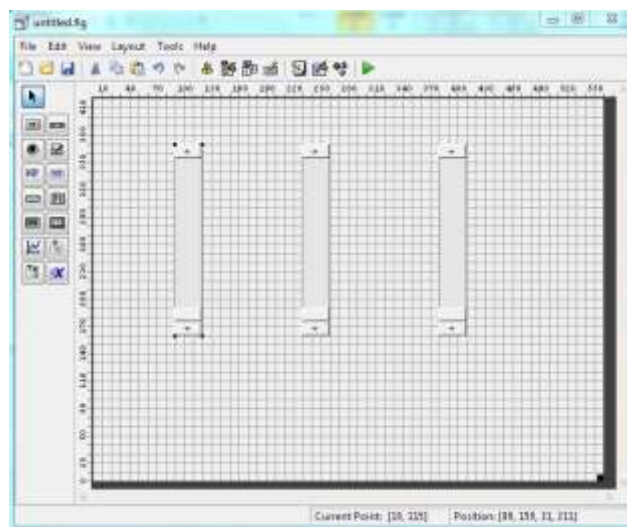
1. Ketikkan *guide* pada *command window*

Sehingga akan tampil kotak dialog berikut. Pilih *Blank GUI (Default)* untuk membuat GUI baru.



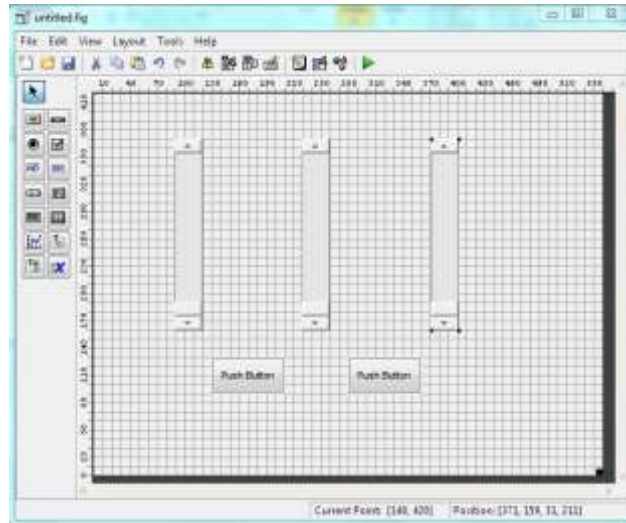
2. Membuat objek-objek yang akan ditampilkan pada GUI

a. Membuat 3 buah slider

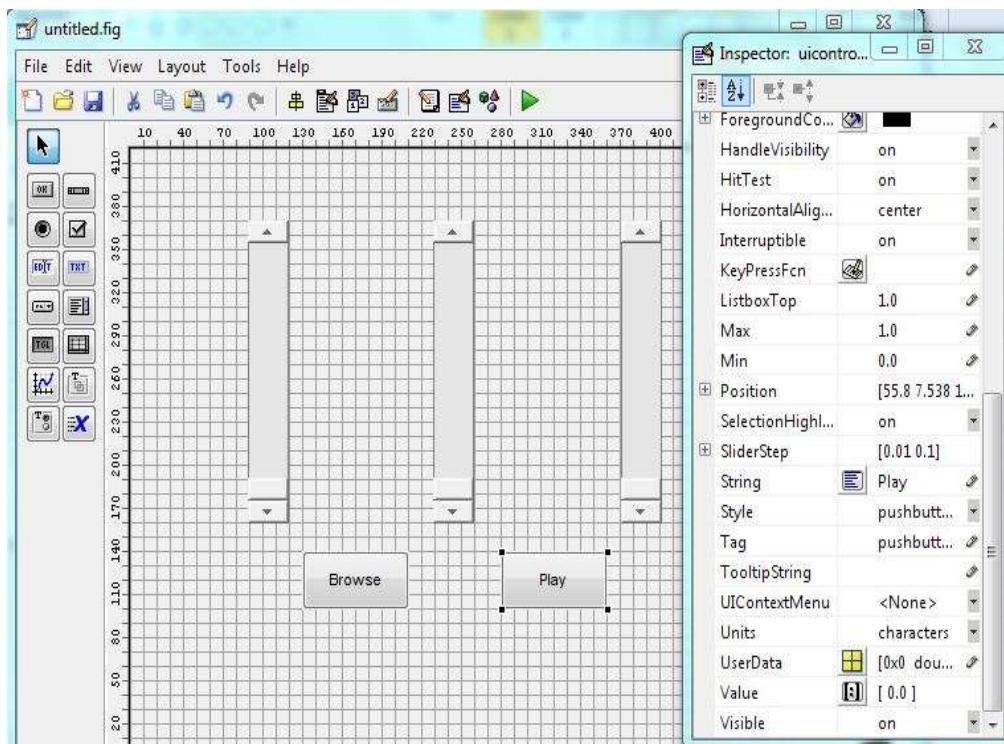


Karena *equalizer* yang dibuat adalah *equalizer 3 band*, maka digunakan 3 objek *slideryang* berfungsi untuk mengatur *gain* yang diinginkan dari setiap kelompok frekuensi (*band*).

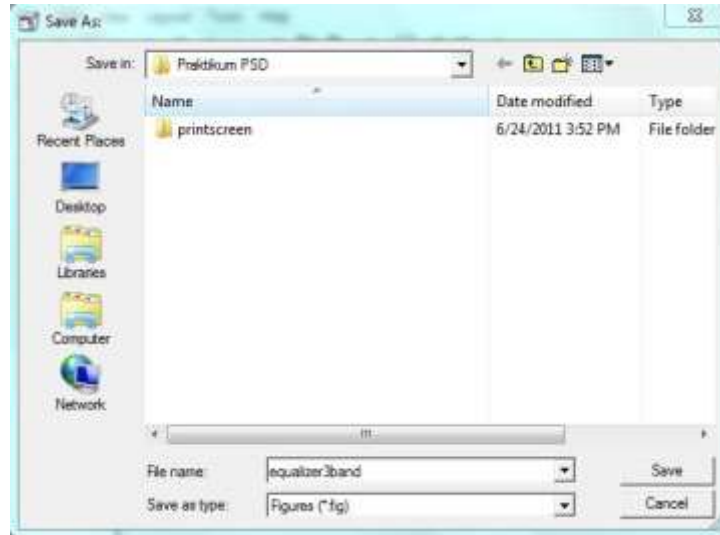
b. Membuat 2 buah push button



Dua buah *push button* yang dibuat adalah tombol *Browse* untuk mengambil file wav dan tombol *Play* untuk memainkan file wav yang telah difilter. Ganti tulisan di kedua *push button* melalui *Property Inspector*.



5. Simpan file tersebut dan beri nama equalizer 3 band

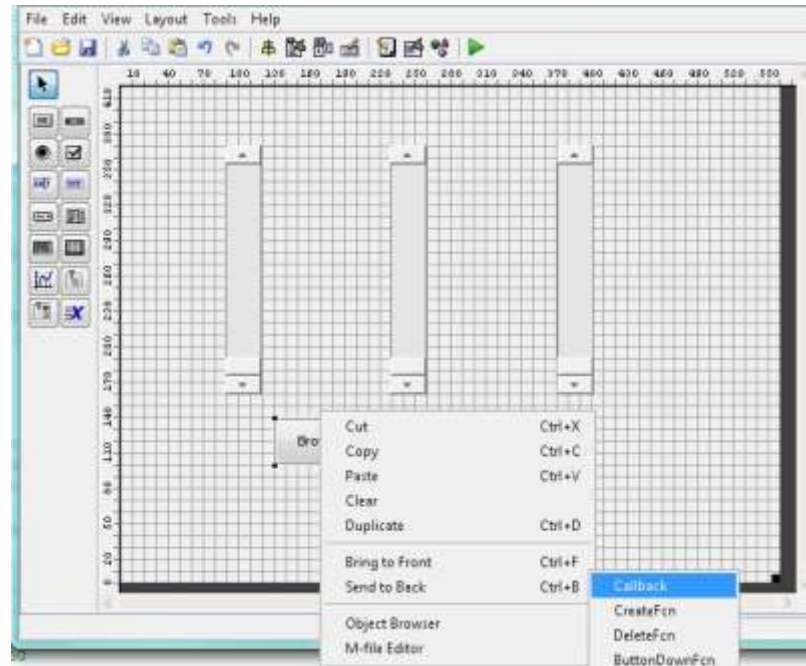


Dengan demikian secara otomatis kita telah membuat 2 file sekaligus, equalizer3band.fig dan equalizer3band.m

6. Membuat *callback function* untuk *pushbutton*

Callback function adalah fungsi yang dipanggil ketika sebuah *push button* diklik. Dengan kata lain, *callback function* inilah yang membuat sebuah tombol (*push button*) mampu melakukan fungsi tertentu ketika diklik.

Klik kanan pada objek *push button* – *View Callbacks*– *Callback*



a. Ketikkan *syntax* berikut untuk *callback function* tombol *Browse*

`% --- Executes on button press in pushbutton1.`

`function pushbutton1_Callback(hObject, eventdata, handles)`

`% hObject handle to pushbutton1 (see GCBO)`

`% eventdata reserved - to be defined in a future version of MATLAB`

`% handles structure with handles and user data (see GUIDATA)`

`% mengambil file dengan format .wav dengan lokasi path`

`[file path]=uigetfile('*.wav','Select a .wav file');`

`% membaca file dari lokasi path dengan nama variabel x dan`

`% frekuensi sampling fs`

`[x fs]=wavread([path file]);`

`save input.mat x fs %menyimpan variable x dan fs dalam file input.mat`



b. Ketikkan *syntax* berikut untuk *callback function* tombol *Play*

% --- Executes on button press in pushbutton2.

function pushbutton2_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

load **input.mat** x fs%memuat kembali (me-load) variabel x dan fs dalam file input.mat

% mengambil nilai dari slider

g1=get(handles.slider1,'value');

g2=get(handles.slider2,'value');

g3=get(handles.slider3,'value');

gain1=10^(g1/10);

gain2=10^(g2/10);

gain3=10^(g3/10);

% membuat filter dengan orde 50

b1=fir1(50,300/(fs/2));%LPF

b2=fir1(50,[300, 3400]/(fs/2));%BPF

b3=fir1(50,3400/(fs/2),'high');;%HPF

% mem-filter file x dan mengalikannya dengan gain dari masing-masing band

x1=gain1*filter(b1,1,x);

x2=gain2*filter(b2,1,x);

x3=gain3*filter(b3,1,x);



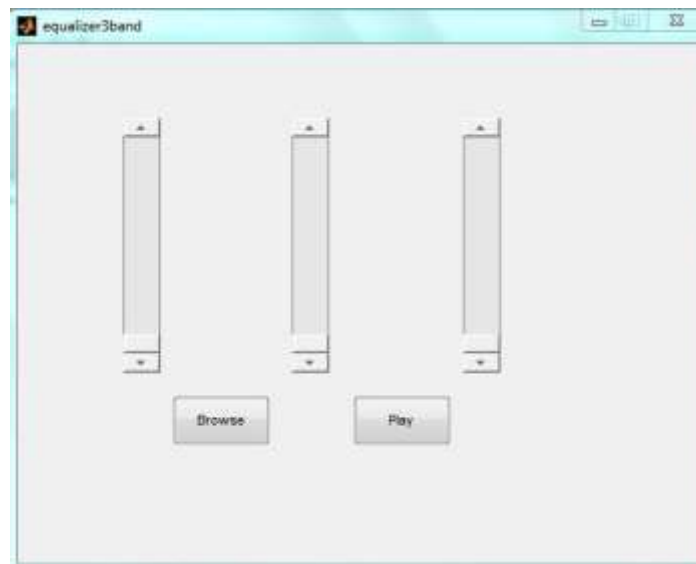
% menggabungkan kembali file-file yang telah difilter

```
x=(x1+x2+x3)/3;
```

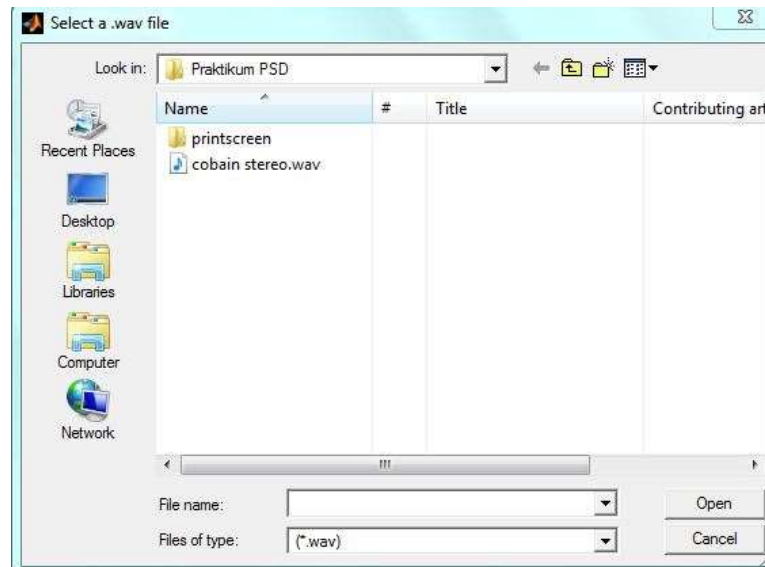
% memainkan file x dengan frekuensi sampling fs

```
wavplay(x,fs)
```

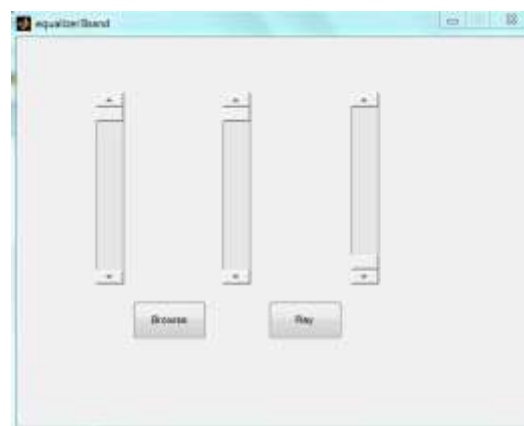
7. Jalankan program yang telah dibuat, maka akan tampil *window* seperti berikut.



- a. Klik tombol *Browse* untuk mengambil file wav yang akan difilter



- b. Naik turunkan posisi *bar* pada ketiga *slider* disesuaikan dengan jenis filter yang diinginkan. Contoh untuk LPF adalah sebagai berikut.



- c. Mainkan file wav yang telah filter dengan menekan tombol *Play*.
Program *equalizer 3 band* di atas adalah contoh aplikasi filter FIR.