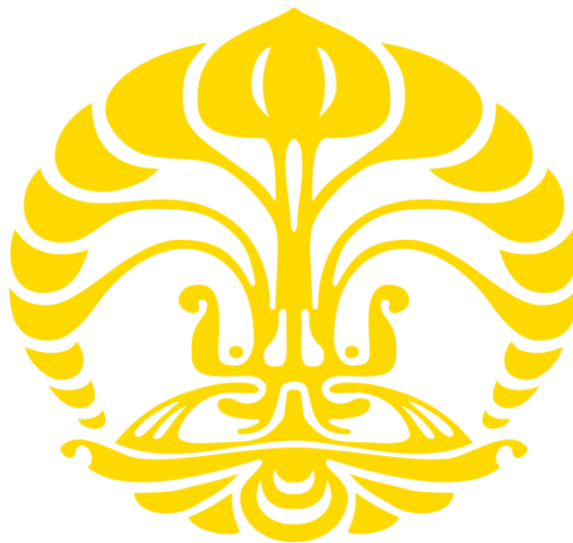


LAPORAN PROYEK AKHIR
SINYAL DAN SISTEM
PROYEK 13: BUTTERWORTH FILTER

Dosen Pengampu :
Prof. Dr. Ir. Dadang Gunawan, M.Eng.



Disusun Oleh :

Hilda Auliana	1906382315
Nidya Anifa	1906382183

PROGRAM STUDI S1 TEKNIK BIOMEDIK
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS INDONESIA
DEPOK

2021

KATA PENGANTAR

Segala puji syukur kami haturkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmatnya kami dapat menyelesaikan laporan proyek 13 yang berjudul “Butterworth Filter” tepat pada waktunya demi memenuhi tugas/proyek akhir Mata Kuliah Sinyal dan Sistem.

Kami menyadari bahwa dalam penulisan buku ini tidak terlepas dari bantuan banyak pihak, terutama Prof. Dr. Ir. Dadang Gunawan, M.Eng., selaku dosen Mata Kuliah Sinyal dan Sistem yang telah memberikan materi serta membimbing kami dalam pembuatan laporan ini. Maka dari itu kami mengucapkan terima kasih sebesar-besarnya.

Kami juga menyadari bahwa di dalam hasil laporan ini terdapat banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, kami mohon maaf dan sangat mengharapkan kritik serta saran yang membangun demi menyempurnakan kembali laporan ini.

Penyusun

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI	3
1. BAB I : PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Dasar Teori	4
2. BAB II : PEMBAHASAN	6
2.1 Soal	6
2.2 Deskripsi Program	6
3. BAB 3 : PENUTUP	17
3.1 Kesimpulan	17
REFERENSI	18

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sinyal merupakan besaran fisik yang didalamnya mengandung informasi dan dapat direpresentasikan secara matematis dengan fungsi yang memiliki satu atau lebih variabel. Dalam kehidupan sehari-hari kita banyak menemui bermacam sinyal, contohnya sinyal suara. Meski begitu pada kenyataannya sinyal-sinyal tersebut tidak sepenuhnya terbebas dari *noise*. *Noise* merupakan bagian yang tidak diinginkan dari sinyal yang dapat menginterferensi informasi yang terkandung pada sinyal. Untuk menghilangkan atau mengurangi *noise* dapat dilakukan dengan melakukan pemrosesan sinyal berupa *filtering*.

Filter adalah suatu program untuk mengambil sinyal pada rentang frekuensi tertentu dan menghilangkan atau melemahkan sinyal dengan frekuensi diluar frekuensi tersebut. Dengan filter yang tepat maka *noise* yang berada pada frekuensi tertentu dapat dihilangkan sehingga didapatkan sinyal yang diinginkan dan meningkatkan kualitas dari sinyal tersebut. Pada proyek ini dilakukan pembuatan program yang dapat memfilter sinyal sinusoidal dan sinyal audio dengan menggunakan Matlab.

1.2 Tujuan

Terdapat beberapa tujuan utama dalam pembuatan proyek ini yang telah terangkum sebagai berikut:

1. Memahami pengertian dari filter
2. Memahami cara kerja Butterworth Filter
3. Membuat Butterworth Filter pada input sinyal sinusoidal dan audio dengan matlab
4. Mengetahui pengaruh filter pada suatu sinyal

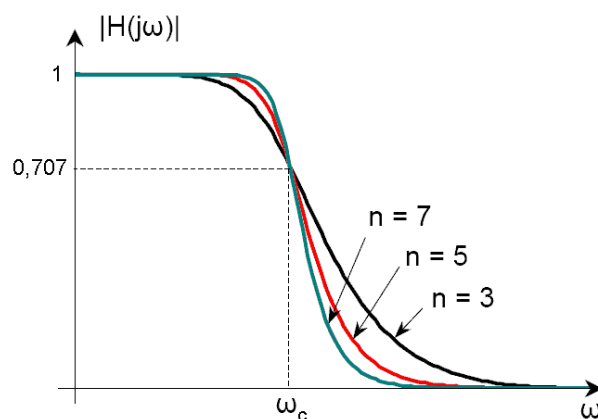
1.3 Dasar Teori

Filter digital merupakan sebuah filter dalam sistem waktu diskrit yang digunakan untuk mengurangi serta meningkatkan aspek tertentu dari sinyal tersebut. Filter ini memiliki kelebihan dalam performanya yang tinggi dengan *transition zone* yang lebih kecil, ketahanan, serta fleksibilitas dalam menentukan range dibandingkan filter analog. Salah satu tipe filter analog yang sering digunakan dalam aplikasi *Digital Signal Processing* (DSP) yaitu filter *Infinite Impulse Response* (IIR). Filter IIR memiliki respon terhadap impulse satuan (*unit impulse*) dengan panjang tak terhingga. Filter tersebut membutuhkan koefisien yang lebih

sedikit untuk respon frekuensi yang curam sehingga dapat mengurangi jumlah waktu komputasi. Salah satu jenis filter IIR yang sering digunakan yaitu filter Butterworth. Filter Butterworth atau dapat disebut juga filter magnitudo rata maksimal sebab memiliki *roll-off* paling tajam dalam orde tertentu yang memungkinkan tidak menimbulkan ripple pada respon frekuensi yang terbaca dalam Bode Plotter. Dalam mencapai kerataannya, filter ini mengorbankan daerah transisi yang relatif lebar dari *passband* ke *stopband*, dengan karakteristik transien rata-rata. Konfigurasi ini akan menekan aproksimasi karakteristik *low pass* dengan hasil respons yang mendekati titik nol dengan halus dan rata. Respon frekuensi tersebut dapat didefinisikan dalam formula berikut:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

di mana ω_c merupakan frekuensi *cutoff* filter dan n merupakan orde filter. Dalam frekuensi yang lebih rendah akan didapatkan penguatan yang mendekati 1 dan seiring meningkatnya frekuensi, maka penguatan tersebut juga akan ikut berkurang. Transisi ini bergantung pada orde filter yang digunakan, semakin kecil orde, maka akan semakin mulus *roll-off* yang tercapai, begitu pula sebaliknya. Ketika orde yang digunakan semakin besar, maka akan lebih terlihat seperti step function dengan transisi yang sangat tajam, dimana pada frekuensi yang lebih tinggi akan dihasilkan penguatan yang sangat rendah sehingga dapat terlihat seperti grafik berikut ini.



BAB II PEMBAHASAN

2.1 Soal

Write a MATLAB program to generate a signal vector with 200 samples which consists of a unit sine wave at 5 KHz plus another sine wave with amplitude 10 at 15 KHz, assuming a time step of 0.025 ms. You should design a Butterworth digital filter with cutoff at 10 KHz and power gain down at least 60 dB at 15 KHz. Your program should be able to design the filter with various parameters values.

2.2 Deskripsi Program

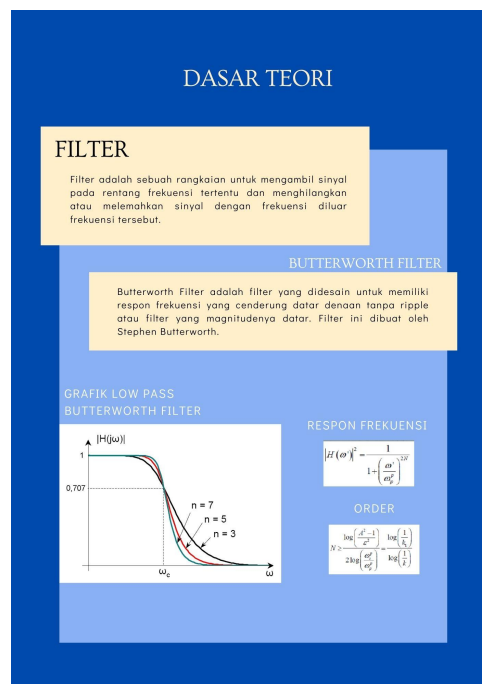
Program ini merupakan program untuk membuat *butterworth filter* pada sinyal sinusoidal dan sinyal audio. Program dibuat pada Matlab R2019b dengan GUI yang didesain pada *App Designer*. Dengan begitu program ini dapat diekspor menjadi sebuah aplikasi sehingga untuk pengoperasiannya lebih fleksibel tanpa memerlukan matlab versi tertentu. Untuk memudahkan penggunaan digunakan tab sehingga user dapat berpindah halaman dengan lebih mudah.

2.2.1 Cover



Pada tab ini menampilkan anggota tim yang membuat program ini serta data diri singkat berupa nama, NPM, jurusan, dan angkatan. Tampilan tab ini dibuat dengan memasukkan gambar yang sudah didesain terlebih dahulu.

2.2.2 Dasar Teori



Pada tab ini terdapat dasar teori dari filter dan butterworth filter yang digunakan sebagai dasar pembuatan program. Tampilan tab ini dibuat dengan memasukkan gambar yang sudah didesain terlebih dahulu.

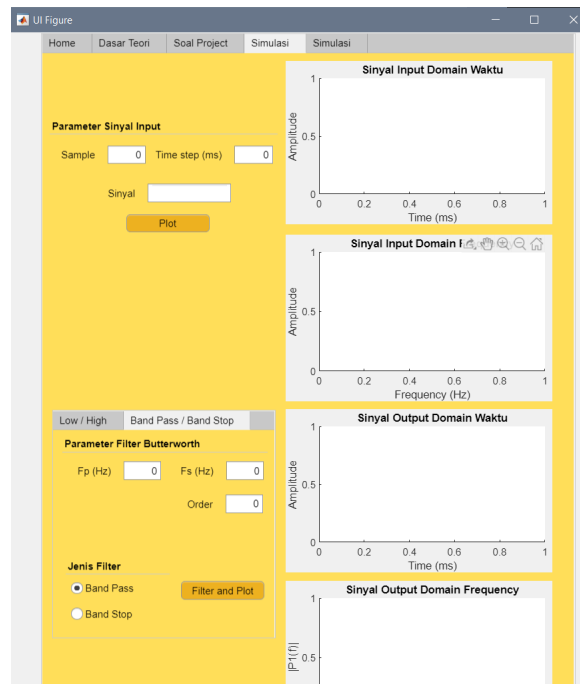
2.2.3 Soal Project

? Problem

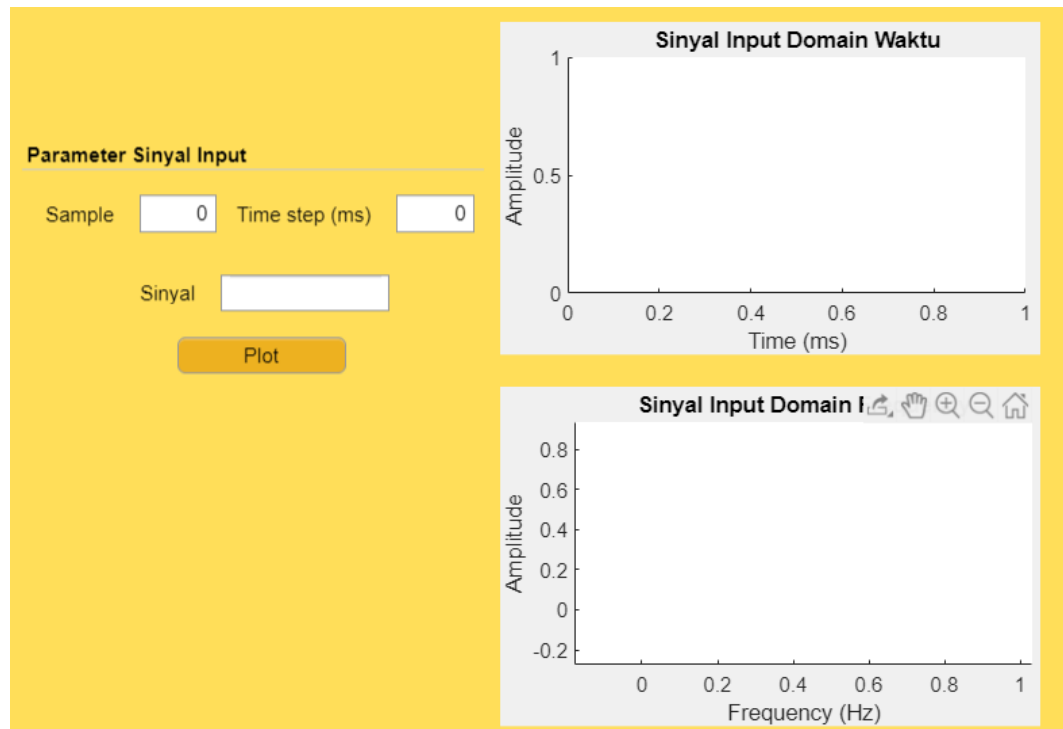
Write a MATLAB program to generate a signal vector with 200 samples, which consists of a unit sine wave at 5 KHz plus another sine wave with amplitude 10 at 15 KHz, assuming a time step of 0.025 ms. You should design a Butterworth digital filter with cutoff at 10 KHz and power gain down at least 60 dB at 15 KHz. Your program should be able to design the filter with various parameters values

Pada tab ini dituliskan soal proyek 13 yang diminta untuk dibuat programnya. Tampilan tab ini dibuat dengan memasukkan gambar yang sudah didesain terlebih dahulu.

2.2.4 Simulasi Sinyal Sinus



Tab ini merupakan untuk mensimulasikan penerapan filter pada sinyal sinus. Terdapat tiga bagian utama dari tab ini yaitu parameter sinyal input, parameter filter butterworth, dan grafik sinyal input maupun output.




```

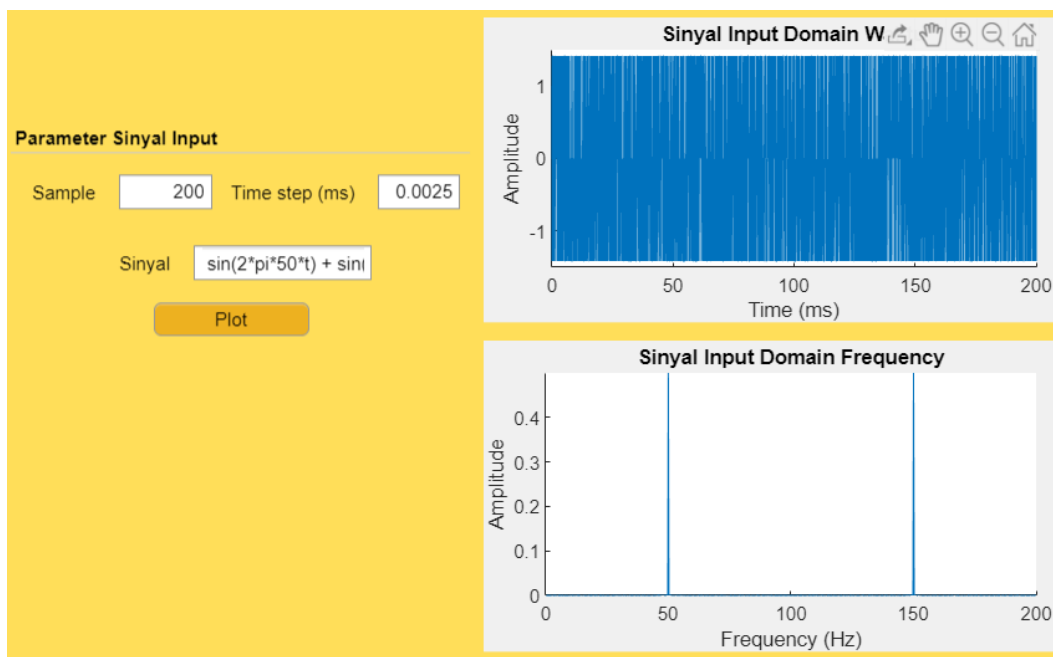
function PlotTimeInputPushed(app, event)
    clc
    close all

    T = app.Sample.Value;
    Tss = app.Timestep.Value;
    t = 0:Tss:T;
    L = length(t);
    n = 2^nextpow2(L);
    Fs = 1/Tss;

    x = eval(app.Sinyal.Value);
    plot(app.SinyalInputAxes, t, x);

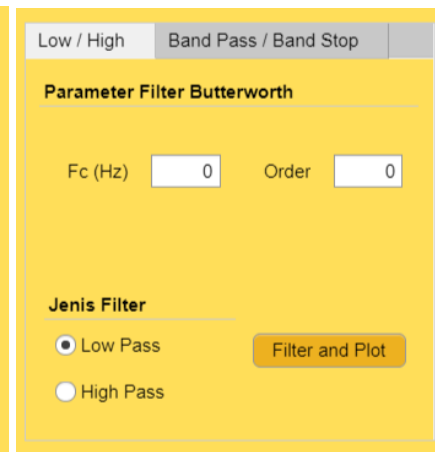
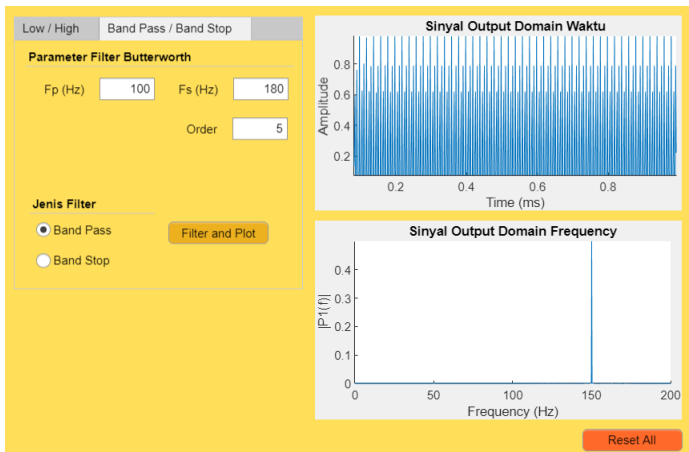
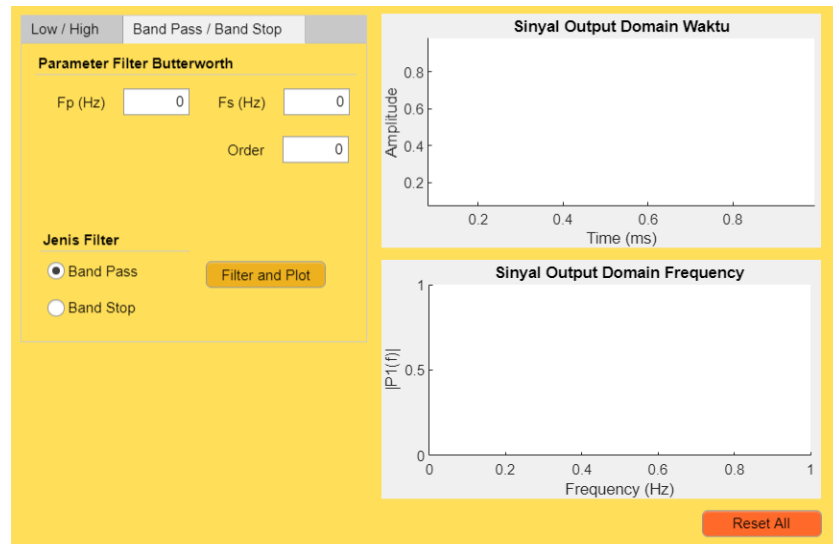
    %untuk FFT
    Y = fft(x, n);
    P2 = abs(Y/L);
    f = Fs*(0:(n/2))/n;
    plot(app.fftInputAxes, f, P2(1:n/2+1));
end

```



Pada bagian ini user dapat memasukkan parameter sinyal input yang diinginkan. Parameter yang dapat dimasukkan yaitu sampel, time steps, dan persamaan sinyal itu sendiri. Ketika user memencet button plot maka akan dihasilkan grafik sinyal input pada domain waktu dan domain frekuensi. Grafik pada domain frekuensi dibuat dengan melakukan FFT (*Fast Fourier Transform*) pada sinyal input sehingga dapat diketahui besaran frekuensi dari sinyal input.

Untuk percobaan dimasukkan parameter sampel sebanyak 200, *time steps* 0.0025 ms dan sinyal berupa penjumlahan dua sinyal sinusoidal dengan frekuensi yang berbeda yaitu $\sin(2\pi \cdot 50 \cdot t) + \sin(2\pi \cdot 150 \cdot t)$. Ketika dipencet plot maka akan ditampilkan grafik sinyal input pada domain waktu dan domain frekuensi. Dapat terlihat hasil FFT pada plot domain frekuensi dari sinyal tersebut yaitu 50 Hz dan 150 Hz sesuai dengan input yang dimasukkan.



```
function PlotLowHighPushed(app, event)
    T = app.Sample.Value;
    Tss = app.Timestep.Value;
    t = 0:Tss:T;
    L = length(t);
    n = 2^nextpow2(L);
    Fs = 1/Tss;
    x = eval(app.Sinyal.Value);
    plot(app.SinyalInputaxes, t, x);

    %utk high/low pass filter
    fp = app.FpLowHigh.Value;
    N = app.FsLowHigh.Value;

    if (app.LowPass.Value)
        jenis = 'low';
    end

    if (app.HighPass.Value)
        jenis = 'high';
    end

    wn = (fp)*2/Fs;
    [b, a] = butter(N, wn, jenis);

    freqs(b,a)
    title(sprintf('Magnitude and Phase Response'))

    %Filtering
    u = filter(b,a,x);

    %Plot the signal output
    plot(app.Outputaxes, [1:L]/Fs, u);

    %FFT
    Y0 = fft(u, n);
    P20 = abs(Y0/L);
    f0 = Fs*(0:(n/2))/n;
    plot(app.fftOutputAxes, f0, P20(1:n/2+1));
```

```

function PlotTimeBandButtonPushed(app, event)
    T = app.Sample.Value;
    Tss = app.Timestep.Value;
    t = 0:Tss:T;
    L = length(t);
    n = 2^nextpow2(L);
    Fs = 1/Tss;
    x = eval(app.Sinyal.Value);
    plot(app.SinyalInputaxes, t, x);

    %utk band pass/stop filter
    fp1 = (app.FpBand.Value)*2/Fs;
    fp2 = (app.FsBand.Value)*2/Fs;
    wn = [fp1 fp2];
    N = app.OrderBand.Value;

    if (app.BandPass.Value)
        jenis = 'bandpass';
    end

    if (app.BandStop.Value)
        jenis = 'stop';
    end

    %Calculate the filter coefficient
    [b, a]=butter(N, wn, jenis);

    freqs(b,a)
    title(sprintf('Magnitude and Phase Response'))

    %Filtering
    u = filter(b,a,x);

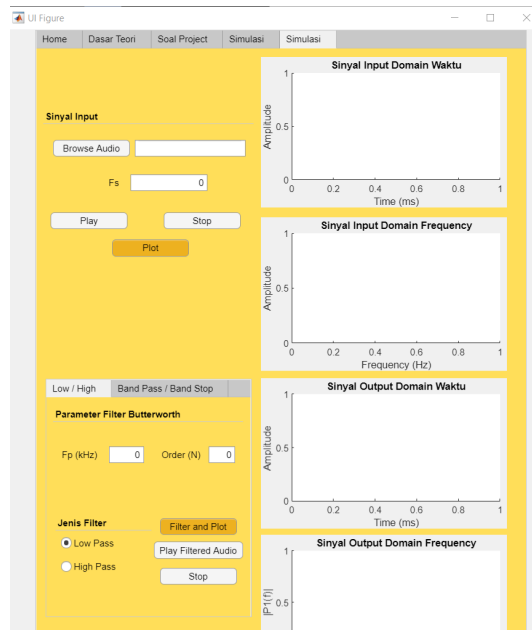
    %Plot the signal output
    plot(app.Outputaxes, [1:L]/Fs, u);

    %FFT
    Y0 = fft(u, n);
    P20 = abs(Y0/L);
    f0 = Fs*(0:(n/2))/n;
    plot(app.fftOutputAxes, f0, P20(1:n/2+1));

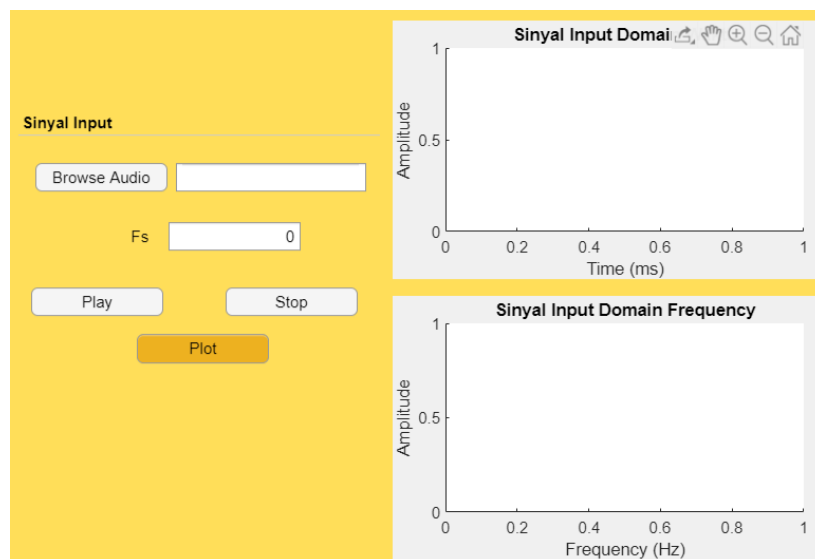
```

Pada bagian ini user dapat memasukkan parameter *filter butterworth*. Terdapat dua tab yang berbeda untuk *lowpass filter/high pass filter* dan *bandpass filter*. Pada tab *low/high* dapat dimasukkan parameter *cut-off frequency* (Hz) dan order filter. User dapat memilih tipe *lowpass filter* atau *highpass filter*. Ketika dipencet *button filter and plot* maka akan dihasilkan grafik sinyal output pada domain waktu dan domain frekuensi serta respon magnitude dan fase filter. Pada tab *pass/stop band* dapat dimasukkan parameter *higher cut-off frequency* (Hz), *lower cut-off frequency* (Hz), dan order filter. User dapat memilih tipe *bandpass filter* atau *stopband filter*. Ketika dipencet *button filter and plot* maka akan dihasilkan grafik sinyal output pada domain waktu dan domain frekuensi serta respon magnitude dan fase filter.

2.2.5 Simulasi Audio File



Tab ini merupakan untuk mensimulasikan penerapan filter pada sinyal audio file yang dapat dimasukkan oleh user. Terdapat tiga bagian utama dari tab ini yaitu parameter sinyal input, parameter filter butterworth, dan grafik sinyal input maupun output.



```
function BrowseAudioButtonPushed(app, event)
    global Fs;
    global Audio;
    global player;

    [filename, pathname] = uigetfile( ...
        { '*.wav', 'WAV File (*.wav)' }, ...
        'Select an Wav File');
    if isequal(filename,0)||isequal(pathname,0)
        app.AudioFileText.Value = 'file not found';
    else
        app.AudioFileText.Value = sprintf(['Selected File : ', filename]);

        [Audio, Fs] = audioread([pathname,filename]);
        player = audioplayer(Audio, Fs);

        app.FsAudio.Value = Fs;
    end
end
```

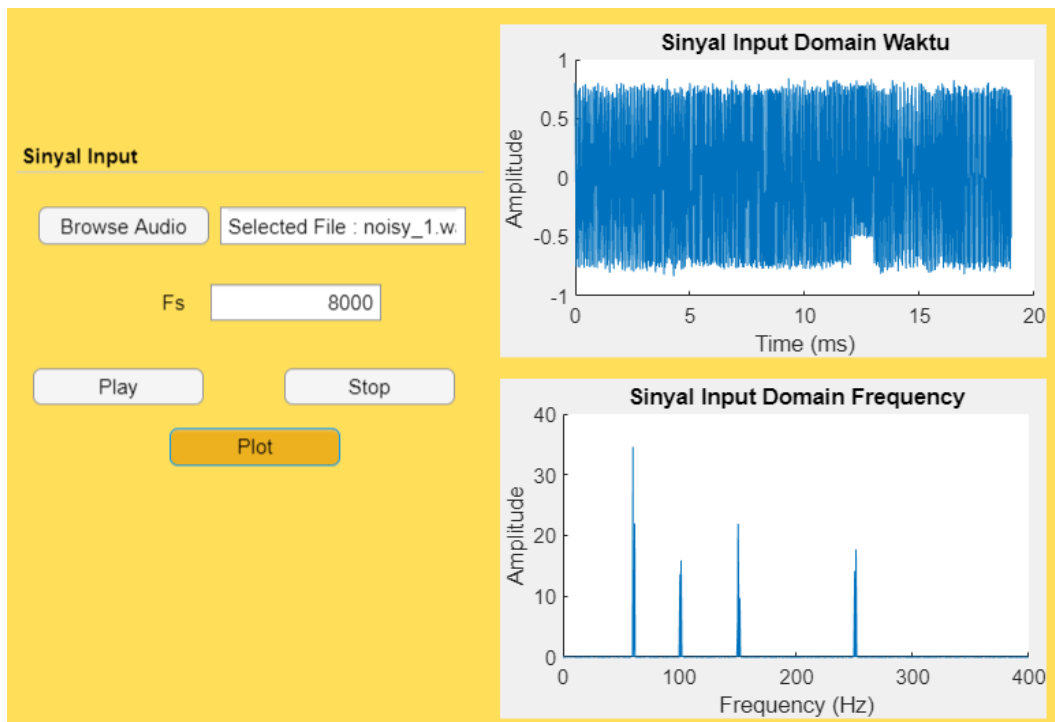
```
function PlotTimeInputAudioButtonPushed(app, event)
    global Audio;
    global Fs;

    t = 0:1/Fs:(length(Audio)-1)/Fs;
    L = length(t);
    n = pow2(nextpow2(L));

    Y = fft(Audio, n);
    P2 = abs(Y).^2/n;
    f = (0:n-1)*(Fs/n)/10;

    plot(app.fftInputAxesAudio, f(1:floor(n/2)),P2(1:floor(n/2)))

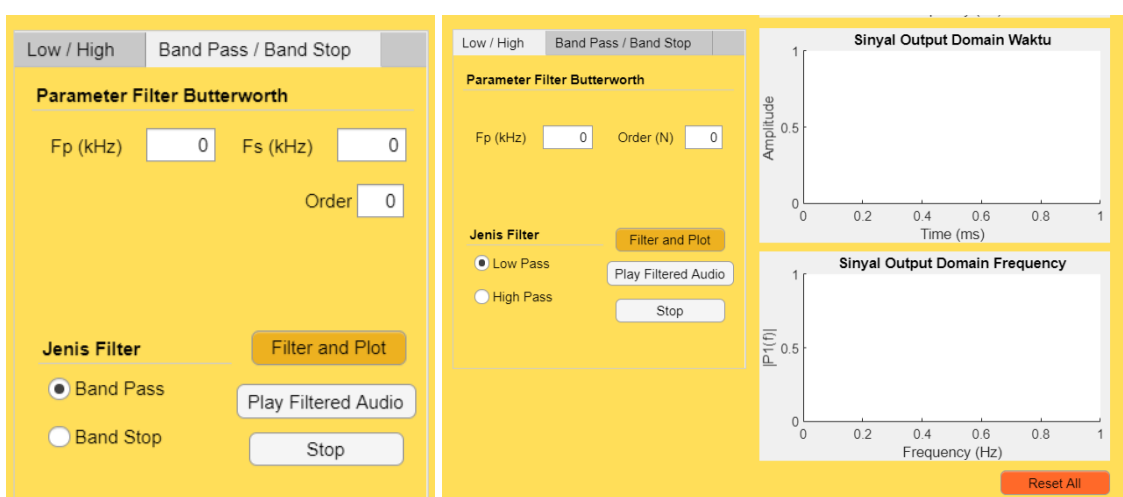
    plot(app.SinyalInputaxesAudio, t, Audio);
end
```

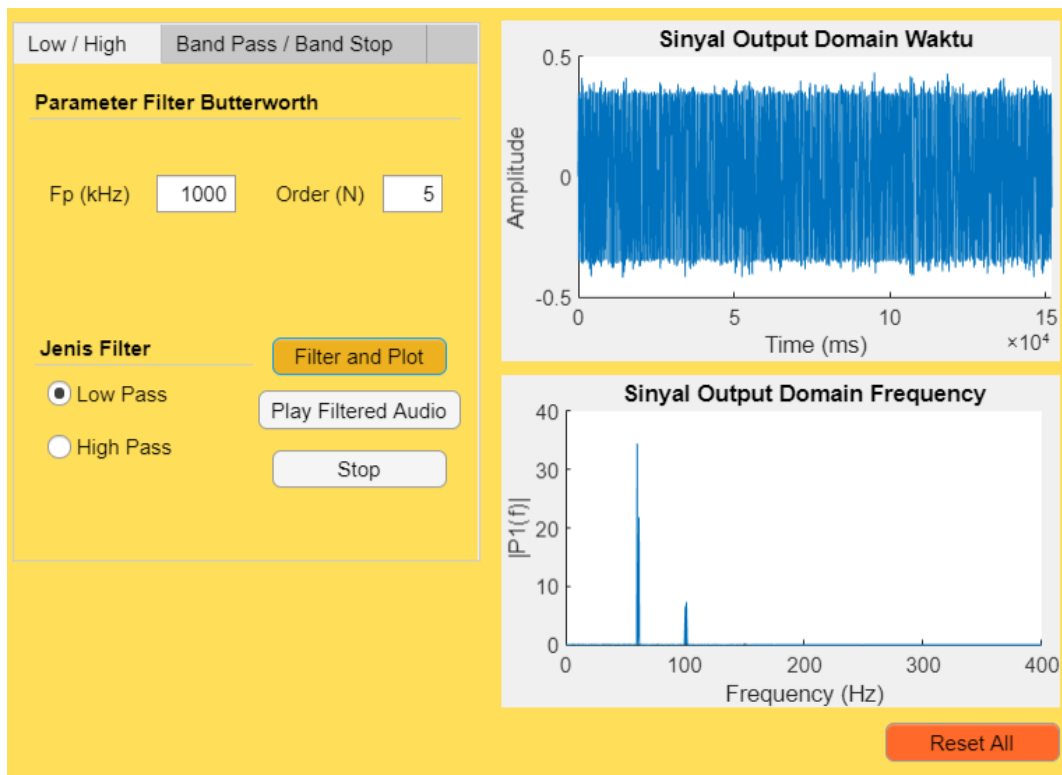


Pada pemrosesan awal sinyal input, audio dengan format .wav dapat dimasukkan dengan menekan Browse Audio. Setiap info dalam audio tersebut kemudian akan terbaca melalui fungsi berikut:

```
[y,Fs] = audioread(filename)
```

Data yang dihasilkan nantinya akan berupa karakteristik sinyal dan frekuensi samplanya (F_s). Selanjutnya, sinyal input akan diproses menggunakan FFT untuk melihat domain frekuensi audio awal tersebut seperti gambar di atas. Selain itu, audio juga dapat diputar melalui tombol Play dan dihentikan juga melalui tombol Stop.





Seperti yang terlihat pada tab berikut, setelah sinyal input yang berupa audio ditampilkan, maka sinyal tersebut kemudian akan diproses kembali menggunakan filter Butterworth seperti dibawah ini.

```
[b,a] = butter(n,Wn,ftype)
```

Dalam fungsi desain filter butterworth, diperlukan beberapa parameter seperti nilai order (N) dan frekuensi *cut-off* yang telah dinormalisasi (wn). Nilai wn akan diperoleh melalui formula berikut:

$$F_p \cdot 2 / F_s$$

dimana F_p merupakan input frekuensi *cut-off* dari *user* dan F_s merupakan frekuensi sample audio. Perbedaan antara *bandpass/bandstop* dan *lowpass/highpass* terletak pada banyaknya frekuensi *cut-off* yang perlu dimasukkan, pada *bandpass/bandstop* dibutuhkan sebanyak 2 *cut-off*, sementara pada *lowpass/highpass* hanya dibutuhkan 1 nilai *cut-off*. Setelah itu, input audio awal akan melalui tahap *digital filtering* melalui fungsi berikut:

```
y = filter(b,a,x)
```

Audio yang telah terfilter kemudian akan diproses kembali pada sistem fft untuk melihat domain frekuensi yang telah terfilter. Pada hasil akhir, maka akan terlihat grafik berupa sinyal output, domain frekuensinya, serta respon frekuensi yang digunakan dalam filter.

```

function PlotLowHighAudioButtonPushed(app, event)
    global Audio;
    global Audiofilt;
    global Fs;
    global player2;

    t = 0:1/Fs:(length(Audio)-1)/Fs;
    L = length(t);
    n = pow2(nextpow2(L));

    %utk high/low pass filter
    fp = app.FpLowHigh_2.Value;
    N = app.OrderLowHigh_2.Value;

    if (app.LowPass_2.Value)
        jenis = 'low';
    end

    if (app.HighPass_2.Value)
        jenis = 'high';
    end

    %Normalized the frequencies
    wn=(fp)*2/Fs;

    %Calculate the filter coefficient
    [b, a] = butter(N,wn, jenis);

    freqs(b,a);
    title(sprintf('Magnitude and Phase Response'))

    %Filtering
    Audiofilt = filter(b,a,Audio);
    player2 = audioplayer(Audiofilt,Fs);

    %Plot the signal output
    plot(app.OutputaxesAudio, Audiofilt);

    %FFT
    Y0 = fft(Audiofilt, n);
    P20 = abs(Y0).^2/n;
    f0 = (0:n-1)*(Fs/n)/10;
    plot(app.fftOutputAxesAudio, f0(1:floor(n/2)),P20(1:floor(n/2)));
end

function PlotBandAudioButtonPushed(app, event)
    global Audio;
    global AudiofiltB;
    global Fs;
    global player2B;

    t = 0:1/Fs:(length(Audio)-1)/Fs;
    L = length(t);
    n = pow2(nextpow2(L));

    %utk band pass/stop filter
    fp1 = (app.FpBand_2.Value)*2/Fs;
    fp2 = (app.FsBand_2.Value)*2/Fs;
    wn = [fp1 fp2];

    N = app.OrderBand_2.Value;

    if (app.BandPass_2.Value)
        jenis = 'bandpass';
    end

    if (app.BandStop_2.Value)
        jenis = 'stop';
    end

    %Calculate the filter coefficient
    [b, a]=butter(N,wn, jenis);

    freqs(b,a)
    title(sprintf('Magnitude and Phase Response'))

    %Filtering
    AudiofiltB = filter(b,a,Audio);
    player2B = audioplayer(AudiofiltB,Fs);

    %Plot the signal output
    plot(app.Outputaxes, AudiofiltB);

    %FFT
    Y0 = fft(AudiofiltB, n);
    P20 = abs(Y0).^2/n;
    f0 = (0:n-1)*(Fs/n)/10;
    plot(app.fftOutputAxesAudio, f0(1:floor(n/2)),P20(1:floor(n/2)));
end

```

BAB III

PENUTUP

3.1 Kesimpulan

1. *Butterworth Filter* dapat digunakan untuk menghilangkan *noise* atau sinyal yang tidak diinginkan pada frekuensi tertentu.
2. Jika suatu sinyal diberi *filter* maka akan menghilangkan sinyal pada frekuensi tertentu dan mempertahankan sebagian yang lain sehingga *noise* yang tidak diinginkan hilang dan kualitas sinyal membaik.
3. Pada *high pass filter* akan meloloskan sinyal yang lebih tinggi dari *cut-off frequency*. Pada *low pass filter* akan meloloskan sinyal yang lebih rendah dari *cut-off frequency*. Pada *bandpass filter* akan meloloskan sinyal yang diantara kedua *cut-off frequency*. Pada *bandstop filter* akan meloloskan sinyal yang diluar kedua *cut-off frequency*.
4. Semakin tinggi order maka sinyal yang terfilter akan lebih presisi

REFERENSI

- [1] "How to filter noisy signal by using IIR filter - MATLAB Answers - MATLAB Central", Mathworks.com. [Online]. Available: <https://www.mathworks.com/matlabcentral/answers/214648-how-to-filter-noisy-signal-by-using-iir-filter>. [Accessed: 04- Jun- 2021].
- [2] "Butterworth filter design - MATLAB butter", Mathworks.com. [Online]. Available: <https://www.mathworks.com/help/signal/ref/butter.html>. [Accessed: 04- Jun- 2021].
- [3] "Fast Fourier transform - MATLAB fft", Mathworks.com. [Online]. Available: https://www.mathworks.com/help/matlab/ref/fft.html?searchHighlight=fft&s_tid=srchtitle. [Accessed: 04- Jun- 2021].
- [4] "Read audio file - MATLAB audioread", Mathworks.com. [Online]. Available: https://www.mathworks.com/help/matlab/ref/audioread.html?s_tid=srchtitle. [Accessed: 04- Jun- 2021].
- [5] "Fourier Transforms- MATLAB & Simulink", Mathworks.com. [Online]. Available: https://www.mathworks.com/help/matlab/math/fourier-transforms.html?searchHighlight=bluewhale&s_tid=srchtitle. [Accessed: 13- Jun- 2021].
- [6] "Declare variables as global - MATLAB global", Mathworks.com. [Online]. Available: <https://www.mathworks.com/help/matlab/ref/global.html>. [Accessed: 14- Jun- 2021].
- [7] "Butterworth Filters - an overview | ScienceDirect Topics", Sciencedirect.com. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/butterworth-filters>. [Accessed: 13- Jun- 2021].
- [8] L. Lidyawati, A. Darlis and A. Tamba, "IMPLEMENTASI FILTER INFINITE IMPULSE RESPONSE (IIR) DENGAN RESPON BUTTERWORTH DAN CHEBYSHEV MENGGUNAKAN DSK TMS320C6713", *Jurnal Elektro dan Telekomunikasi Terapan*, vol. 2, no. 1, 2016. Available: <https://media.neliti.com/media/publications/195478-ID-implementasi-filter-infinite-impulse-res.pdf>. [Accessed: 13-June-2021]

