



UNIVERSITAS INDONESIA

**IMPLEMENTASI *AUDIO EQUALIZER* PADA MATLAB
MENGUNAKAN MANIPULASI ALGORITMA**

Proyek Akhir Pengolahan Sinyal Digital

| | |
|--------------------------------|-------------------|
| RISTI DWI PUTRI | 1506669702 |
| AKMAL DUTA SATRIA | 1506670130 |
| EUFRAAT TSAQIB QASTHARI | 1506740332 |

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
PROGRAM STUDI FISIKA
DEPOK
2018**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena dengan rahmat dan karunia-Nya kami dapat menyelesaikan laporan proyek akhir mengenai “*Implementasi Audio Equalizer pada Matlab menggunakan Manipulasi Algoritma*” dengan baik meskipun masih terdapat kekurangan dalam laporan yang kami buat ini. Laporan proyek akhir ini ditulis dalam rangka memenuhi persyaratan mata kuliah Pengolahan Sinyal Digital pada Departemen Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia.

Penulisan laporan ini, sangat dipengaruhi oleh orang-orang sekitar penulis. Oleh karena itu, penulis mengucapkan terima kasih kepada Bapak Adhi Harmoko Saputro S.Si., M.Kom., Ph.D sebagai dosen mata kuliah Pengolahan Sinyal Digital, orang tua dan keluarga telah yang memberikan semangat, motivasi dalam melakukan proyek akhir ini. Harapan penulis semoga laporan ini dapat berguna dan membantu menambah pengetahuan serta wawasan. Selain itu, penulis juga menyadari bahwa baik dalam segi sistematika penyusunan maupun materi yang dipaparkan masih terdapat banyak kekurangan. Oleh karena itu, penulis berharap agar adanya kritik dan saran yang dapat membantu kami untuk perbaikan di masa yang akan datang.

Jakarta, 25 Desember 2018

Penulis

DAFTAR ISI

| | |
|-------------------------------------|-----------|
| KATA PENGANTAR | .ii |
| DAFTAR ISI | iii |
| BAB 1 PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Perumusan Masalah | 2 |
| 1.3. Tujuan | 2 |
| 1.4. Manfaat | 2 |
| BAB 2 TINJAUAN PUSTAKA | 3 |
| 2.1. Transformasi Fourier | 3 |
| 2.2. Tipe - tipe Filter | 3 |
| 2.2. Filter Butterworth | 4 |
| BAB 3 METODE PENELITIAN | 6 |
| 3.1. Rancangan Sistem | 7 |
| 3.2. Rancangan Pengolahan Algoritma | 9 |
| BAB 4 HASIL DAN PEMBAHASAN | 10 |
| 4.1. Hasil | 10 |
| 4.2. Pembahasan | |
| BAB 5 KESIMPULAN DAN SARAN | |
| 5.1. Kesimpulan | |
| 5.2. Saran | |
| DAFTAR REFERENSI | |

BAB 1

PENDAHULUAN

Bab ini memberikan gambaran umum mengenai penelitian yang dilakukan, seperti latar belakang, perumusan masalah, tujuan, dan manfaat.

1.1 Latar Belakang

Setelah abad ke-20, pengolahan sinyal digital telah berkembang. Dengan munculnya komputer, dan perangkat digital lainnya, mendukung perkembangan dari pemrosesan sinyal digital. Equalization adalah sebuah efek yang memungkinkan penggunaannya untuk mengontrol tanggapan frekuensi dari sinyal keluaran. *Audio equalization* menyediakan berbagai cara untuk mendapatkan suara yang lebih bersih. Pergeseran fasa, kebisingan kuantisasi, dan suara elektromagnetik juga merupakan masalah umum yang muncul selama konversi analog ke digital. Tujuan mengembangkan equalizer adalah untuk membatasi faktor-faktor ini yang memiliki dampak negatif pada kualitas sinyal audio, yang dapat dicapai dengan meningkatkan atau memotong bagian-bagian sinyal[1].

Dalam pengolahan sinyal digital (DSP), aplikasi seperti MATLAB menyediakan berbagai alat yang dapat digunakan untuk *audio equalization*. Alat-alat ini menawarkan cara yang lebih sederhana bagi penggunaannya untuk membuat fungsi, skrip, dan model yang dapat diimplementasikan dalam berbagai topik yang berkaitan dengan bidang studi yang berbeda. MATLAB, seperti banyak aplikasi pengkodean lainnya, berisi beberapa pustaka toolbox yang nyaman dengan algoritma dan fungsi untuk memfilter. Fungsi Butterworth MATLAB digunakan untuk merancang filter untuk equalizer ini.

1.2 Perumusan Masalah

Permasalahan yang akan diangkat pada proyek akhir ini adalah :

1. Bagaimana cara mengimplementasikan *multi-band equalizer* pada matlab menggunakan manipulasi algoritma?
2. Bagaimana audio yang dihasilkan setelah dilakukan *equalization*?

1.3 Tujuan

Tujuan dari pembuatan proyek akhir ini yaitu sebagai berikut:

1. Mengimplementasikan berbagai filter pada proses pengolahan sinyal digital.
2. Mempelajari dan memahami salah satu aplikasi pengolahan sinyal dalam kehidupan sehari-hari.

1.4 Manfaat

Dari pembuatan proyek akhir ini diharapkan mendapatkan audio yang seimbang dan sesuai dengan yang diinginkan.

BAB 2

TINJAUAN PUSTAKA

Bab ini berisi tentang teori-teori penunjang penelitian yang terdiri dari teori dasar mengenai alat dan parameter yang digunakan, serta permasalahan yang dibahas.

2.1 Transformasi Fourier

2.1.1 Pengenalan

Pada konversi analog ke digital, sinyal melalui sampler, quantizer dan coder. Pada sampler terdapat dua karakteristik yang bersangkutan untuk equalization, yaitu periode sampling, T'' dan frekuensi sampling, F'' , dari sinyal. Pada frekuensi sampling sinyal suara yang umumnya digunakan adalah 44100 Hz, untuk dapat menghasilkan frekuensi sinyal yang dapat direkonstruksi sampai dengan 22500 Hz, sehingga rentang dapat terdengar oleh kebanyakan manusia.

Pada kenyataannya, sinyal adalah fungsi waktu, pada domain waktu. Domain waktu adalah representasi dari nilai nyata yang terjadi karena diberikan interval waktu tertentu. Nilai representasi yang lebih baik dapat didapatkan pada domain frekuensi. Domain frekuensi, yang ditemukan oleh Joseph Fourier, menunjukkan bahwa periode sinyal dapat direpresentasikan dengan deret tak hingga dari sinyal frekuensi sinusoidal maupun diskrit. Di matematika, transformasi antara domain waktu dan domain frekuensi adalah transformasi Fourier [2]. Transformasi Fourier penting untuk mengekualisasi suara, pada analisis sinyal pada domain waktu, kemungkinan sukar untuk di observasi properti atau karakteristik penting yang berkaitan dengan sinyal tersebut.

2.1.2 Transformasi Z (*Z-Transformation*)

Mengambil transformasi Fourier pada sinyal menghadirkan

informasi yang lebih bermakna dari sinyal tersebut pada frekuensi tertentu. Kumpulan algoritma digunakan untuk menghitung transformasi Fourier secara diskrit atau dikenal juga dengan Fast Fourier Transform (FFT). Cara lain yang lebih mudah untuk merepresentasi sinyal adalah transformasi Z. Pada pengolahan sinyal digital (*Digital Signal Processing*), transformasi Z adalah cara yang berguna untuk mengetahui stabilitas sinyal dan juga dapat digunakan untuk merepresentasikan data sinyal yang sudah disampling. Transformasi Z berhubungan dengan transformasi Fourier dengan: $z = e^{j\omega}$.

2.1.3 Frekuensi *Cut-off*

Frekuensi *Cut-off*, ω_n , adalah sinyal pembatas. Frekuensi *Cut-off* dapat dibuat untuk mengetahui dimana sinyal harus berhenti atau ter-*Cut-off* pada frekuensi tertentu. Pada semua filter, frekuensi *Cut-off* digunakan pada wilayah dimana sinyal ingin ditingkatkan.

2.1.4 Lebar Pita (*Bandwidth*)

Untuk membatasi efek *equalizer* dari sinyal suara maka harus ditentukan lebar pita atau *bandwidth*. *Bandwidth* adalah rentang frekuensi dimana *equalizer* dapat bekerja. Hal ini penting karena komponen sistem harus menentukan jarak dimana *equalizer* butuh untuk diimplementasikan agar dapat sinyal output dapat ditingkatkan.

2.1.5 Frekuensi Tengah

Frekuensi tengah adalah frekuensi di tengah-tengah antara frekuensi *cut-off* bawah dan *cut-off* atas. Frekuensi tengah penting karena gain yang lebih besar dari lebar pita memiliki maksimum pada frekuensi tengah sinyal audio. Frekuensi tengah dapat digeser agar rentang kecondongan yang diinginkan frekuensi berada pada satu lebar pita [2].

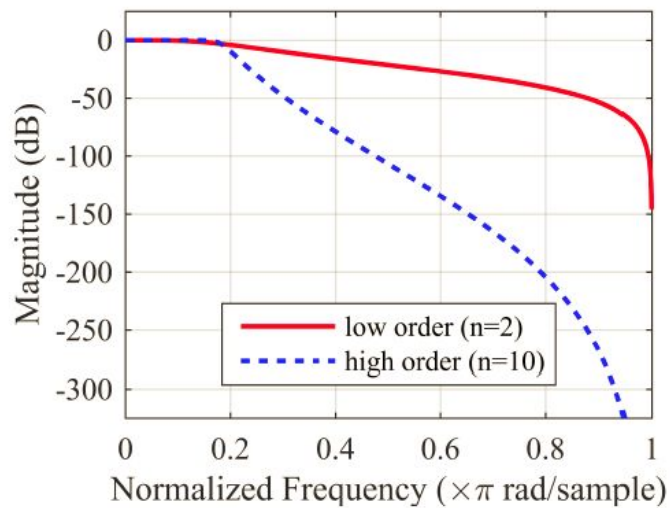
2.2 Tipe-Tipe Filter

2.2.1 Pengenalan

Umumnya ada empat filter yang digunakan untuk *equalization* suara yaitu *lowpass*, *bandpass*, *bandstop* dan *highpass*. Filter-filter ini adalah filter yang digunakan untuk memperbolehkan rentang-rentang mana saja yang sesuai dengan filter itu sendiri. Untuk kebutuhan riset, dapat digunakan semua filter kecuali *bandstop*, sehingga hanya tiga yang dibahas[3].

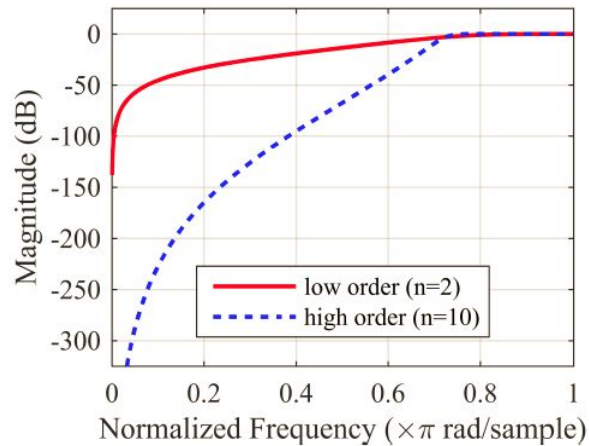
2.2.2 Filter *Lowpass*

Filter *lowpass* hanya memperbolehkan frekuensi bawah yang melewati filter dan menolak frekuensi tinggi, Apabila frekuensi lebih rendah dari frekuensi *cutoff*, filter *lowpass* akan memperbolehkan frekuensi tersebut. Alternatifnya, apabila frekuensi lebih besar dari frekuensi *cutoff*, filter ini akan menolak frekuensi tersebut. Hal tersebut dapat digambarkan dengan grafik dibawah ini:



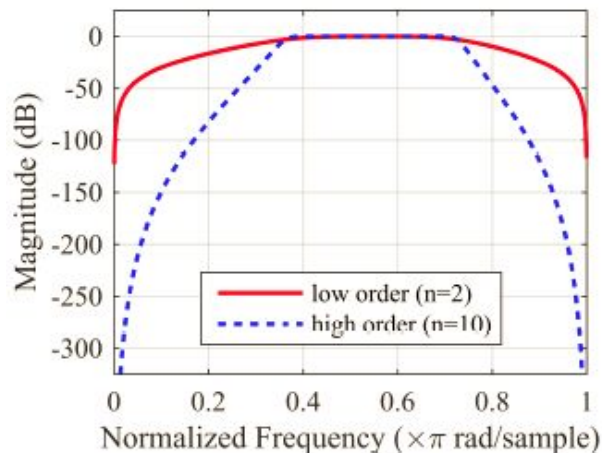
2.2.3 Filter *Highpass*

Filter *high-pass* bekerja dengan memperbolehkan frekuensi yang lebih tinggi dari frekuensi *cut-off* dan menolak frekuensi yang lebih rendah dari frekuensi *cutoff*. Filter ini bekerja berkebalikan dari filter *lowpass* [3]. Grafik dibawah menggambarkan bagaimana filter *high-pass* bekerja:



2.2.4 Filter *Bandpass*

Filter bandpass adalah filter yang menggabungkan cara kerja filter *low pass* dan filter *high pass*, dengan menggunakan frekuensi *cut-off* tinggi dan rendah. Filter ini digunakan di tengah-tengah dari spektrum frekuensi yang ada untuk menghasil rentang lebar pita tertentu. Grafik dibawah ini menunjukkan bagaimana filter *bandpass* bekerja:



2.3 Filter *Butterworth*

Filter *butterworth* adalah fungsi di MATLAB yang menggunakan lima langkah algoritma untuk memfilter sebuah sinyal. Filter ini bekerja pertama kali

dengan menemukan pole-pole analog *low pass*, *zeros* dan *gain* menggunakan fungsi 'buttap'. Apabila atribut-atributnya sudah ditemukan maka akan dikonversi menjadi bentuk *state-space*, dimana transformasi di aplikasikan untuk merubah filter *lowpass* satu ke filter yang lainnya. Pada desain filter digital, filter analog dikonversikan dengan transformasi biliner lalu representasi *state-space* dirubah kembali ke fungsi transfernya [4]. Untuk menjalankan *syntax* Butterworth, koefisien vektor [b, a] adalah nilai yang dihasilkan berdasarkan dari fungsi transfernya, fungsi dibawah adalah fungsi filter Butterworth:

$$H(z) = \frac{B(z)}{A(z)}$$

Parameter input dari *syntax* Butterworth adalah desain filter dengan order, n, frekuensi *Cut-off*, ω_n , dan satu dari empat filter seperti yang telah disebutkan diatas. Filter *low pass* dan filter *high pass* menggunakan satu element vektor untuk frekuensi cutoff. Tetapi, bandpass dan bandstop filter membutuhkan vektor dua elemen karena filter ini membutuhkan dua batas. Berdasarkan dengan teorema Nyquist, atau frekuensi terlipat, ω_n harus diantara 0 dan 1 dengan nilai $\pi \text{ rad/sample}$ dari desain filter. Faktor kualitas dari filter bandpass, atau biasa disebut dengan faktor Q adalah rasio dari tengah-tengah frekuensi ke lebar pitanya. Faktor Q digambarkan dengan bentuk grafik yang diambil dari koefisien-koefisien dari fungsi transfer. Nilai Q yang lebih kecil menghasilkan respon yang lebih lebar dan nilai Q yang besar maka responnya akan semakin rapat [5].

BAB 3

METODE PENELITIAN

3.1 Rancangan Sistem

Desain *audio equalizer* dimulai dengan mendapatkan sinyal input dan mendapatkan nilai untuk setiap filter. Penting untuk dicatat bahwa MATLAB menggunakan besaran ternormalisasi dan bukannya nilai desibel sehingga parameter input ini harus dikonversi. Untuk membangun filter untuk penyamaan sinyal, fungsi Butterworth di MATLAB digunakan bersama filter digital dan amplifier. Desain filter Butterworth memungkinkan pengguna untuk dengan mudah menyesuaikan jumlah band dan parameter filter yang diperlukan untuk tujuan desain tertentu. Desain *audio equalizer* ini menggunakan jenis filter ‘lowpass’, ‘highpass’, dan tiga ‘bandpass’ untuk menyamakan sinyal input. Frekuensi cutoff yang digunakan selama implementasi ditunjukkan pada Tabel 1.1

| Butterworth Filter Type | Cutoff Frequency (Hz) | Testing Gain (dB) | Quality Factor |
|-------------------------|-----------------------|-------------------|----------------|
| Lowpass | 250 | 1 | --- |
| Bandpass | [250 500] | 4 | 1.5 |
| Bandpass | [500 2000] | 4 | 0.833 |
| Bandpass | [2000 4000] | 2 | 1.5 |
| Highpass | 4000 | 1 | --- |

Tabel.1 range frekuensi untuk desain *audio equalizer*

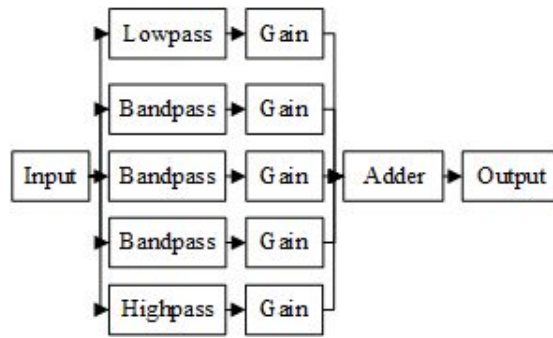
Menggunakan MATLAB, vektor yang berisi data sampel dari input dibuat pada frekuensi sampling tertentu 44100 Hz. Transformasi Fourier dari sinyal input dilakukan untuk merencanakan respon magnitudo normalisasi sinyal dengan frekuensi yang sesuai. Urutan sistem lima dipilih untuk setiap filter karena menyediakan desain filter yang akurat dengan memasukkan cukup banyak input sampel sebelumnya. Termasuk desain yang lebih tinggi akan membuat waktu komputasi menjadi jauh lebih besar dan tidak akan membuat dampak signifikan pada peningkatan keakuratan filter.

3.2.1 Rancangan Pengolahan Algoritma



Gambar 3.1 Flowchart Algoritma dari Audio Equalization

Menggunakan fungsi MATLAB, 'audioread', vektor yang berisi data sampel dari input dibuat pada frekuensi sampling yang ditentukan 44100 Hz. Transformasi Fourier dari sinyal input dilakukan untuk merencanakan respon magnitudo normalisasi sinyal dengan frekuensi yang sesuai. Urutan sistem lima dipilih untuk setiap filter karena menyediakan desain filter yang akurat dengan memasukkan cukup banyak input sampel sebelumnya. Perintah desain yang lebih tinggi akan meningkatkan waktu komputasi dan tidak akan membuat dampak signifikan pada peningkatan keakuratan marginal dari filter.



Gambar 3.2 Blok Diagram *Audio Equalizer*

Pada blok diagram diatas ditunjukkan sinyal input dibagi menjadi enam jalur paralel. Setiap jalur digabungkan dalam seri dengan penguat gain untuk meningkatkan atau memotong sinyal dengan jumlah gain yang diinginkan. Nilai gain direferensikan sebagai variabel dalam kode MATLAB, dan dapat diubah ke nilai yang diinginkan. Jalur keenam dalam model ini berfungsi sebagai jalur kesatuan maju untuk merekonstruksi bagian-bagian sinyal yang tidak diubah oleh filter. Setelah direkonstruksi, sinyal output ditulis ke ruang kerja dan vektor dibuat dengan data yang difilter pada frekuensi sampling. Data ini kemudian dapat ditulis dan dimainkan sebagai sinyal audio untuk pengujian verifikasi.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil

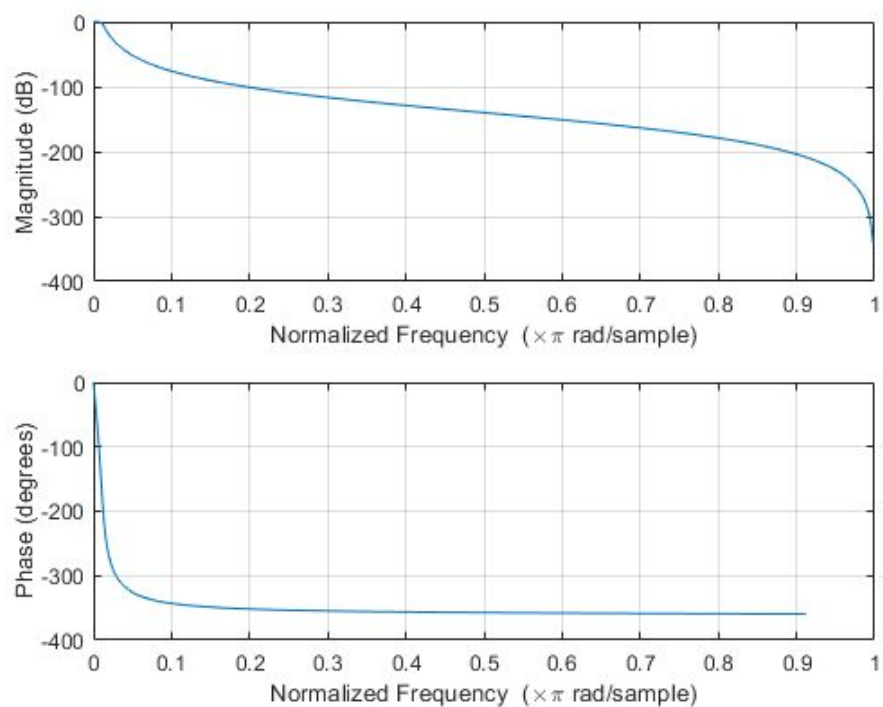
Untuk menentukan nilai n atau order dari filter tersebut digunakan perhitungan dengan mempertimbangkan tipe dari filter itu sendiri dan juga besar atenuasi yang diinginkan. Nilai order ini akan berpengaruh terhadap efek *ripple* pada output apabila mendekati nilai ideal dan juga lambatnya respons dari filter tersebut.

Pada filter pertama yaitu lowpass dan *butterworth* filter, untuk mendapatkan atenuasi sebesar 20dB pada rentang frekuensi cutoff 250Hz didapatkan nilai n sebesar 3,18 yang berarti order n -th dari filter tersebut adalah empat (dinaikkan nilainya sampai integer terdekat).

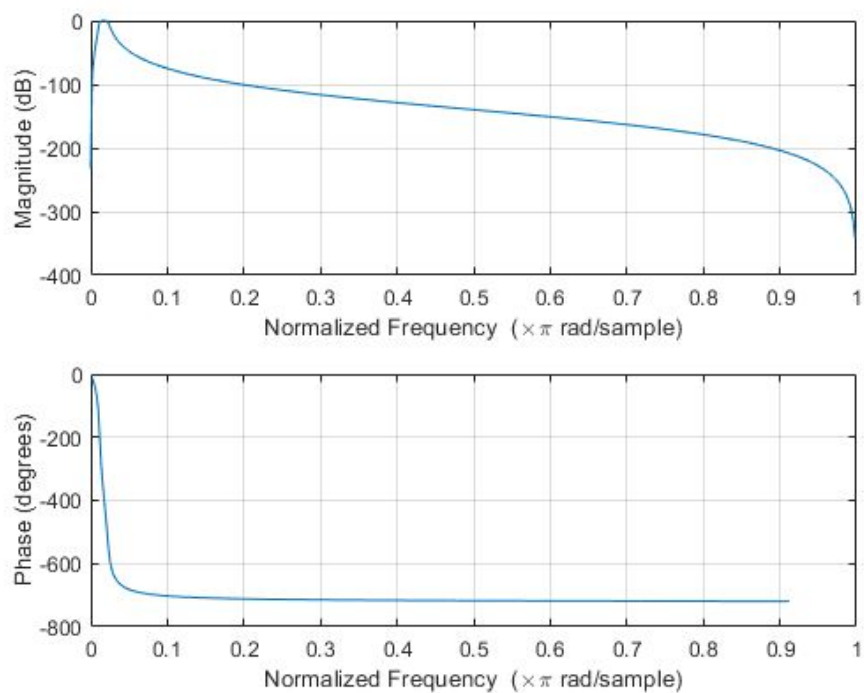
Pada filter kedua yaitu filter bandpass di dapatkan nilai n dengan perhitungan bahwa nilai atenuasi stop band sebesar 5db dan pass band sebesar 25dB maka (jarak atenuasi sebesar 20dB) didapatkan nilai n sebesar 3,59 yang berarti order n -th dari filter tersebut adalah empat pula. Filter bandpass yang lainnya didapatkan nilai order n -th yang sama yaitu empat.

Pada filter ketiga yaitu filter highpass digunakan dengan mengganti persamaan yang menggunakan w/w_c dengan $-w/w_c$ sehingga akan didapatkan nilai order n -th yaitu empat pula.

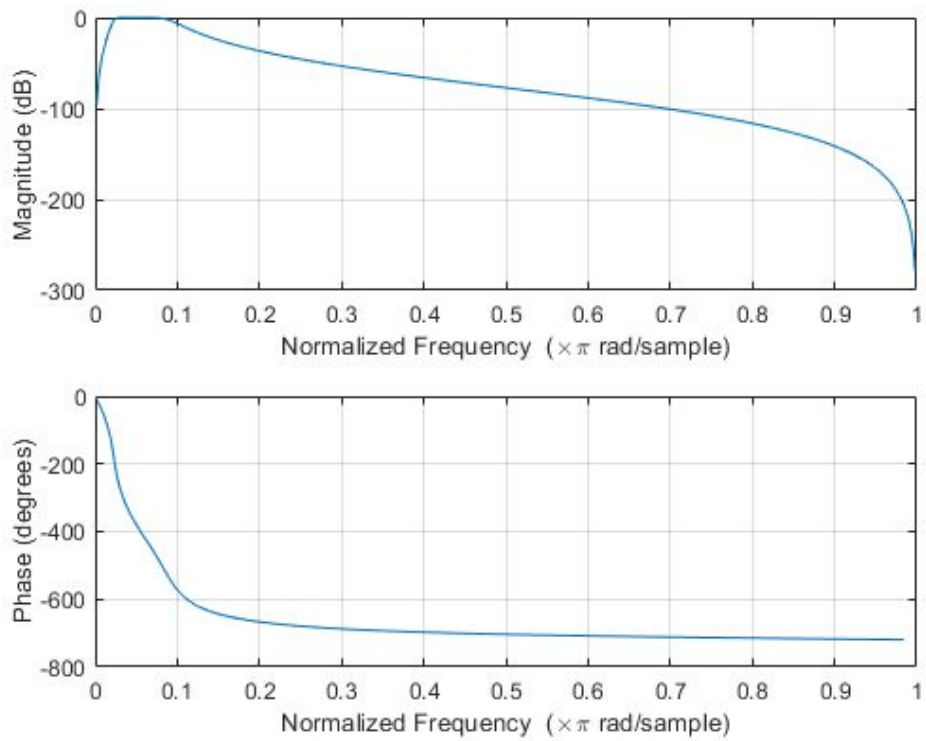
Dengan syarat-syarat diatas, apabila fase (degree) dan besar atenuasinya di grafikkan maka akan dihasilkan grafik filter seperti berikut:



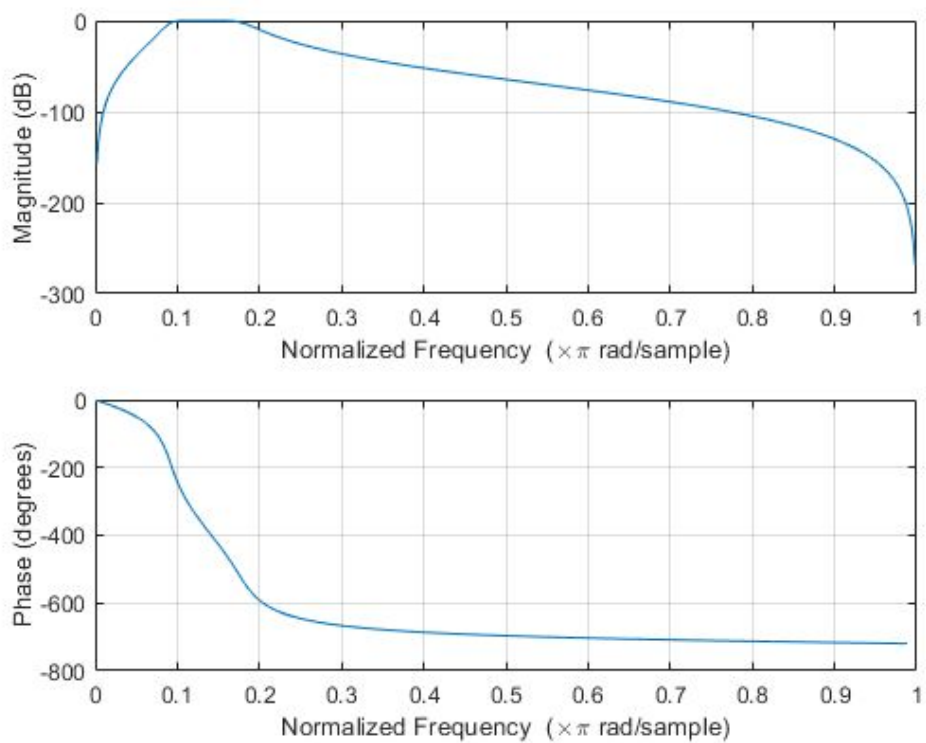
Gambar X. Grafik filter pertama yaitu besar atenuasi (dB) terhadap frekuensi ternormalisasi pada filter lowpass butterworth.



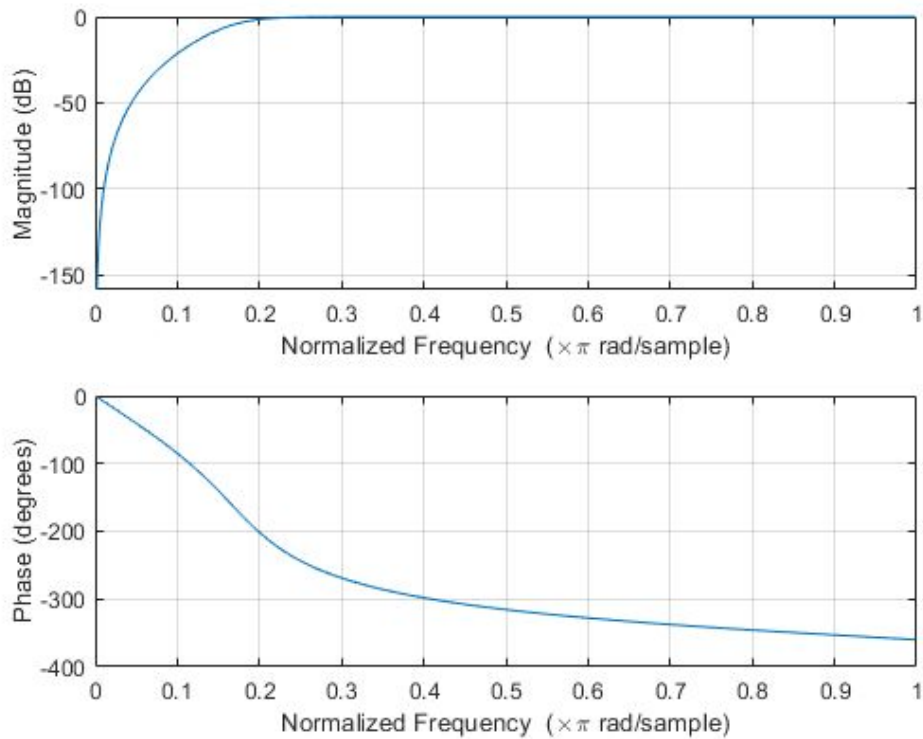
Gambar X. Grafik besar filter kedua yaitu besar atenuasi (dB) terhadap frekuensi ternormalisasi pada filter bandpass butterworth.



Gambar X. Grafik besar filter ketiga yaitu besar atenuasi (dB) terhadap frekuensi ternormalisasi pada filter bandpass butterworth.

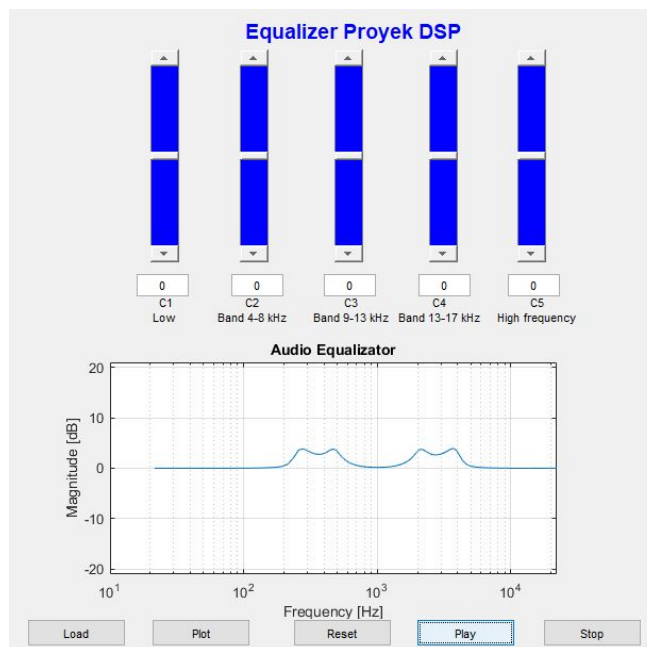


Gambar X. Grafik besar filter keempat yaitu besar atenuasi (dB) terhadap frekuensi ternormalisasi pada filter bandpass butterworth.



Gambar X. Grafik besar filter kelima yaitu besar atenuasi (dB) terhadap frekuensi ternormalisasi pada filter bandpass butterworth.

Untuk menghasilkan suara yang sudah modifikasi suaranya maka digunakan GUI (*Graphical User Interface*) untuk menginput besar atenuasi pada hasil dari gabungan filter-filter diatas akan menjadi satu filter yang dapat dirubah nilai atenuasinya dengan menggunakan *slider* pada rentang frekuensi tertentu seperti contoh pada dibawah ini:



Gambar X. GUI (Graphical User Interface) dari program dengan menampilkan 5 *slider* ekualisasi dan satu buah grafik filter *equalizer*.

4.2 Pembahasan

Pada filter-filter aktif yang dapat digunakan sebagai *low-pass*, *band-pass* dan *high-pass* dapat digunakan berbagai macam filter seperti filter *bessel*, *butterworth* dan *chebysev* yang menggunakan respons *pole*. Sedangkan untuk filter yang menggunakan respons *pole* dan *zero* dapat digunakan *inverse chebysev* dan *elliptic*. Pada desain filter aktif umumnya filter dengan menggunakan respons *zero* akan lebih kompleks, oleh karena itu kami memilih filter yang lebih sederhana seperti *butterworth*, *chebysev* atau *bessel*.

Filter *bessel*, *butterworth* dan *chebyshev* memiliki kelebihan dan kekurangan masing-masing. Pada filter *bessel*, ia memiliki sifat yang lebih baik *time domain* nya yang berarti artifak-artifak *ringing* (sinyal terosilasi yang muncul apabila diberikan sinyal input yang tidak terosilasi) jauh lebih sedikit. Sedangkan filter *chebysev* memiliki *cut-off frequency* yang lebih tajam dibanding keduanya. Oleh karena itu, mengapa kami memilih untuk menggunakan filter *butterworth* karena ia memiliki sifat diantara keduanya (*bessel* dan *chebyshev*) sehingga relatif lebih baik apabila digunakan sebagai filter pada *equalizer*.

Pada proyek ini digunakan pula *filter-order* yang bernilai empat sesuai dengan perhitungan agar desain filter baik. Berarti, nilai order ini dapat mencegah terjadinya *ringing* pada sinyal output yang dikarenakan nilai terlalu tinggi atau nilai yang terlalu rendah yang dapat menyebabkan sinyal output yang kurang responsif.

Dari grafik low-pass filter dapat dilihat bahwa terjadi penurunan dari atenuasi setelah 0π rad sehingga pada 1 lebih dari -300dB berkurang besar atau *magnitude* dari sinyal tersebut. Hal ini, menunjukkan bahwa low-pass filter bekerja untuk mengurangi *magnitude* dari sinyal yang memiliki frekuensi tidak rendah dan meneruskan sinyal yang memiliki frekuensi rendah (*low-pass*).

Grafik band-pass filter dapat dilihat bahwa terjadi penurunan *magnitude* pada rentang-rentang tertentu yang berada pada rentang yang teratuenasi. Filter-filter band-pass hanya meneruskan sinyal pada rentang tertentu saja. Hal ini dapat dilihat dengan grafik filter kedua, ketiga, dan keempat.

Dapat dilihat pula, pada grafik high-pass filter terjadi peningkatan seperti fungsi logaritmik yang berkesesuaian dengan peningkatan frekuensi. Sehingga apa bila frekuensi semakin meningkat, maka nilai *magnitude* yang dihasilkan akan semakin meningkat pula. Pada grafik ini hanya frekuensi-frekuensi tinggi saja yang sinyalnya diperbolehkan teratuenasi.

Nilai-nilai dari filter tersebut akan diatur kembali oleh pengguna, sesuai dengan kebutuhan pengguna. Misalnya, apabila suara terlalu bising dan perlu dikurangi suara-suara tajam, maka pengguna dapat menggunakan GUI yang sudah disediakan untuk mengurangi *magnitude* dari frekuensi-frekuensi tinggi dan sebaliknya apabila pengguna membutuhkan suara-suara yang lebih tinggi diakibatkan oleh ketidak detailannya suatu sinyal pada frekuensi tersebut.s

Setelah ditentukan frekuensi yang terekualisasi maka plot filter akan berubah dan menyesuaikan dengan *slider* mana yang mengalami perubahan. Penggabungan dari filter-filter low-pass, band-pass dan high-pass akan ditampilkan pada satu plot yang secara *default* akan menampilkan 4 *spike* yang diakibatkan oleh filter band-pass.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada proyek ini dapat ditunjukkan bagaimana cara kerja dari sebuah program *equalizer* beserta filter-filter yang ada pada *equalizer* tersebut. Berikut merupakan poin-poin kesimpulan yang dapat kami ambil dari proyek ini:

- Equalizer dapat bekerja dengan menggunakan filter-filter pemrosesan sinyal seperti *bessel*, *butterworth* dan *chebysev*.
- Filter *butterworth* dipilih karena sistemnya yang sederhana dan *cut-off frequency* yang tidak begitu tajam apabila dibandingkan dengan filter lainnya.
- Untuk mendapatkan hasil ekualisasi yang optimal, diperlukan perhitungan orde filter agar tidak terjadi artifak-artifak *ringing* pada sinyal dan mempercepat respons sinyal.
- Filter *low-pass* bekerja dengan melewatkan sinyal dengan frekuensi rendah dan mengecilkan sinyal lainnya.
- Filter *low-pass* bekerja dengan melewatkan sinyal dengan frekuensi tertentu dan mengecilkan sinyal lainnya.
- Filter *low-pass* bekerja dengan melewatkan sinyal dengan frekuensi tinggi dan mengecilkan sinyal lainnya.
- Equalizer bekerja dengan mengatur slider lalu program akan menggabungkan nilai-nilai filter sehingga didapatkan filter utama pada sinyal suara.

5.2 Saran

Berikut merupakan saran-saran yang ditemukan peneliti agar pembaca dapat melanjutkan atau mendiskusikan penelitian ini:

- Apakah perbandingan grafik dari filter-filter lainnya dapat ditunjukkan beserta analisisnya?

- Bagaimana cara kerja filter utama agar dapat menampilkan grafik secara *real-time*? sehingga apabila terjadi perubahan pada *slider* grafik dapat berubah secara langsung.
- Apakah input dari program ini dapat berupa *hardware*?
- Berapa nilai spesifik frekuensinya pada titik-titik terjadi perubahan atenuasi pada sinyal?
- Apakah pemilihan filter dapat dilakukan secara GUI? agar pengguna tidak usah lagi merubah source code untuk mengaplikasikan filter sendiri.

DAFTAR REFERENSI

- [1] Caruso, W. C., Fisk, B. C., Whaley, J. D., Alsaedi, S., & McPherson, B. D. Implementation of a Multi-band Equalizer in MATLAB and Simulink Using Algorithm Manipulation.
- [2] Dick Blanford, John Parr. (2013). Introduction to Digital Signal Processing. First Edition. Upper Saddle River: Pearson,Print.
- [3] College of Saint Benedict & Saint Johns University. [online] Tersedia di : http://www.physics.csbsju.edu/217/digital_filter.pdf [Diakses 21 November 2018]
- [4] Mathworks. [online] Tersedia di: <http://www.mathworks.com/help/signal/ref/butter.html> [diakses 21 November 2018]
- [5] Richard, Oliver and Jean-Marc Jot. (2015). Efficient Multi-Band Digital Audio Graphic Equalizer with Accurate Frequency Response Control, AES 139th Convention.