

Pole-Zero Placement untuk Bass-Boost Audio Filter

Vanny Alviolani Indriyani (13221020)
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
13221020 @mahasiswa.itb.ac.id

Abstrak—Pemfilteran pada sinyal bertujuan untuk menghilangkan sinyal yang tidak diinginkan seperti noise atau sinyal dengan frekuensi tertentu. Low pass filter yang digabungkan dengan penguatan tertentu dapat menghasilkan bass booster untuk sinyal audio. Filter ini meredam sinyal dengan frekuensi di atas frekuensi bass dan melakukan penguatan agar suara bass terdengar lebih keras. Bass booster dapat dibuat menggunakan filter FIR dan IIR. Hasil yang didapatkan yaitu filter IIR dapat melakukan filtering lebih baik dan lebih efisien dibanding filter FIR karena filter IIR menggunakan feedback dari sinyal output untuk dikembalikan ke input. Filter IIR dapat dibuat menggunakan MATLAB maupun menggunakan perhitungan tangan. Filter yang dibuat dengan MATLAB dapat melakukan filtering dengan lebih baik karena koefisien filter yang dihasilkan akurat.

Kata Kunci—Bass-booster, FIR, IIR, bilinear transform.

I. PENDAHULUAN

SELURUH sinyal pada kehidupan manusia tidak ada yang terlepas dari noise atau gangguan baik secara alami maupun buatan. Oleh karena itu, untuk mengolah suatu sinyal diperlukan filter untuk menghilangkan noise yang tidak diinginkan. Filter juga dapat digunakan untuk memanipulasi sinyal sehingga dihasilkan sinyal yang diinginkan. Contoh kegiatan memanipulasi sinyal adalah penggunaan equalizer pada sinyal audio. Equalizer dapat mengisolasi sinyal dengan frekuensi tertentu dan menguatkan sinyal lainnya. Pada penelitian ini dibuat filter yang melakukan bass boost atau mengeraskan bass pada sinyal audio. Filter bass boost dibuat menggunakan filter FIR dan IIR bertipe low pass.

II. LANDASAN TEORETIS

A. Finite Impulse Response (FIR) Filter

Filter FIR adalah filter dengan durasi atau respons impuls yang terbatas, yakni bernilai nol setelah waktu diskrit tertentu. Filter FIR digital dengan orde N memiliki durasi respons impuls dari waktu diskrit $n = 0$ hingga $n = N$ (terdiri dari $N + 1$ sampel/koefisien). Ketentuan orde filter ini digunakan pada berbagai software komputasi seperti MATLAB dan OCTAVE. Perlu diperhatikan bahwa terdapat beberapa referensi yang

menganggap bahwa filter FIR digital dengan orde M memiliki durasi respons impuls dari waktu diskrit $n = 0$ hingga $n = M - 1$ (terdiri dari M sampel). Sistem filter FIR digital dengan orde N dinyatakan dengan persamaan berikut [1].

$$y[n] = b_0x[n] + b_1x[n-1] + \dots + b_Nx[n-N] = \sum_{k=0}^N b_kx[n-k]$$

B. Infinite Impulse Response (IIR) Filter

Filter IIR (Infinite Impulse Response) adalah jenis filter digital yang memiliki respon impuls yang tidak berhenti atau berlangsung selamanya. Filter IIR menggunakan feedback, yaitu output filter dipengaruhi oleh input dan output sebelumnya. Filter IIR dapat menghasilkan respon frekuensi yang kompleks dengan jumlah koefisien yang relatif sedikit, sehingga lebih efisien secara komputasinya. Selain itu Filter IIR juga memiliki koefisien filter dan memori yang relatif sedikit untuk respons frekuensi yang kompleks sehingga dapat mengefisienkan komputasi saat proses filtering. Filter IIR juga memiliki latensi yang rendah sehingga cocok untuk kontrol secara real time dan untuk aplikasi RF berkecepatan tinggi. Kemudian filter IIR juga dapat digunakan untuk meniru karakteristik filter analog menggunakan transformasi pemetaan bidang s - z . Namun, Filter IIR juga memiliki beberapa kelemahan, seperti fase yang tidak linear, stabilitas yang tidak terjamin, dan sensitivitas terhadap kesalahan pembulatan. Sistem filter FIR digital dengan orde N dinyatakan dengan persamaan berikut [1].

$$a_0y[n] = b_0x[n] + b_1x[n-1] + \dots + b_Nx[n-N] - a_1y[n-1] + \dots + a_Ny[n-N] = \sum_{k=0}^N b_kx[n-k] - \sum_{k=1}^N a_ky[n-k]$$

C. Desain Filter IIR dengan Transformasi Bilinear

Transformasi Bilinear adalah hubungan matematis yang dapat digunakan untuk mengubah fungsi transfer filter tertentu di domain kompleks Laplace menjadi domain- z , dan sebaliknya. Filter yang dihasilkan akan memiliki karakteristik yang sama dengan filter aslinya, namun dapat diimplementasikan menggunakan teknik yang berbeda. Transformasi bilinear dapat digunakan untuk menghasilkan potongan respon magnitudo konstan yang mendekati respon magnitudo dari filter analog yang ekuivalen [2].

III. HASIL DAN ANALISIS

Pada penelitian ini, lagu yang difilter adalah lagu berjudul “Hey Mama” yang dipopulerkan oleh David Guetta featuring Nicki Minaj dan Bebe Rexha. Berikut merupakan spektrum frekuensi untuk lagu tersebut.

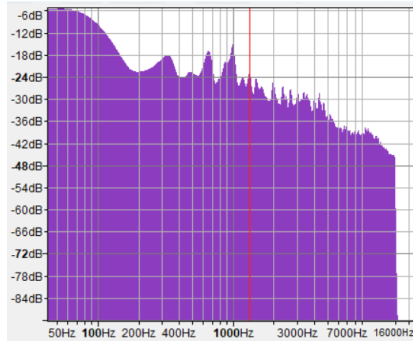


Figure III-1 Spektrum Frekuensi Audio Original

Pada lagu ini, terdapat dua vokalis yang memiliki suara dengan frekuensi yang berbeda yaitu sekitar 300 Hz untuk Nicki Minaj dan 750 Hz untuk Bebe Rexha. Perbedaan frekuensi ini menyebabkan pelemahan pada suara vokal keduanya tidak bisa terjadi secara maksimal karena pelemahan akan berfokus pada salah satu vokal dengan frekuensi tertentu sehingga pelemahan suara vokal lainnya kurang maksimal.

A. Filter FIR MATLAB

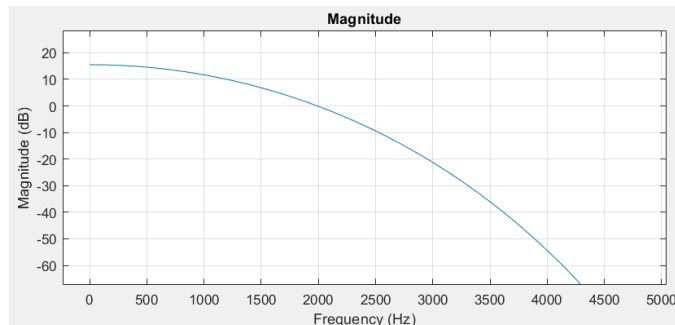


Figure III-2 Respons Frekuensi Filter FIR MATLAB

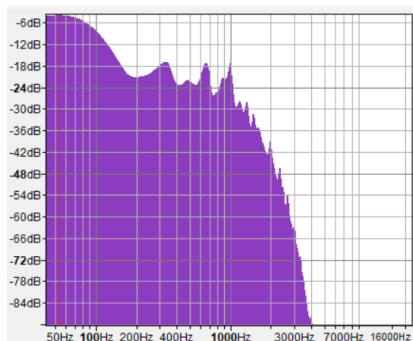


Figure III-3 Spektrum Frekuensi Audio Output Filter FIR MATLAB

```
fir_coeff = 4.1602e-08,2.3954e-07,9.6492e-07,3.1892e-06,9.176e-06,2.3755e-05,5.648e-05,0.00012505,0.00026031,0.0005132,0.00096363,0.0017308,0.0029845,0.0049549,0.0079394,0.012303,0.018471,0.026906,0.038075,0.052402,0.07021,0.091652,0.11665,0.144
```

```
86,0.17559,0.20787,0.24043,0.27178,0.30034,0.32453,0.34296,0.3545,0.35843,0.3545,0.34296,0.32453,0.30034,0.27178,0.24043,0.20787,0.17559,0.14486,0.11665,0.091652,0.07021,0.052402,0.038075,0.026906,0.018471,0.012303,0.0079394,0.0049549,0.0029845,0.0017308,0.00096363,0.0005132,0.00026031,0.00012505,5.648e-05,2.3755e-05,9.176e-06,3.1892e-06,9.6492e-07,2.3954e-07,4.1602e-08
```

Filter FIR yang dibuat pada MATLAB menggunakan orde 16 sesuai dengan spesifikasi. Akan tetapi, dengan jumlah orde ini, atenuasi pada fstop masih terlalu kecil sehingga vokal suara kurang teredam. Oleh karena itu, agar slope filter menjadi lebih curam dan atenuasi pada fstop membesar, orde filter harus diperbesar. Cara memperbesar orde filter adalah dengan melakukan konvolusi antar filter FIR yang telah dibuat. Konvolusi dapat dilakukan berkali-kali sesuai kebutuhan dan setiap konvolusinya akan menggandakan orde filter yang dikonvolusi. Pada penelitian ini, dilakukan konvolusi sebanyak dua kali pada filter FIR berorde 16 sehingga orde akhir filter adalah 64. Meskipun orde filter yang tinggi memiliki slope yang curam, orde filter pada penelitian ini tidak boleh dibuat terlalu besar karena orde yang terlalu besar akan membuat atenuasi pada frekuensi di atas fstop terlalu besar, yaitu bisa mencapai -40 dB pada frekuensi 125 Hz. Pada atenuasi ini, suara vokal akan tidak terdengar karena magnitude nya sudah terlalu kecil. Selain itu, orde filter FIR yang terlalu besar juga merusak sinyal audio karena setelah dilakukan percobaan, filter yang memiliki orde yang lebih tinggi, audio nya lebih terdistorsi atau pecah. Hal ini disebabkan karena orde yang tinggi memiliki respons fasa yang lebih tidak beraturan, sehingga sinyal output nya tergeser-geser dan menjadi terdistorsi.

Koefisien filter yang dihasilkan filter ini jumlahnya ada banyak karena filter ini memiliki orde yang besar. Semakin besar jumlah koefisien, sinyal output akan terdefinisi lebih baik karena sampel rekonstruksi sinyal nya akan lebih banyak. Koefisien filter FIR berjumlah banyak juga karena pada filter ini tidak ada feedback sehingga sinyal output hanya akan bergantung pada sinyal input dan koefisien filter.

Filter ini diatur agar memiliki penguatan sebesar 15.5 dB yaitu dengan cara mengalikan koefisien filter yang telah dikonvolusi dengan $10^{15.5/20}$. Pengalihan ini dilakukan karena secara default, magnitude yang dimiliki oleh filter adalah nol, sehingga agar muncul bass boost atau penguatan pada bass, magnitude audio perlu dikalikan dengan koefisien penguatan. Angka 15.5 dB dipilih karena dianggap sudah cukup untuk menghasilkan suara bass yang boosted. Selain itu, penguatan di atas angka tersebut akan menghasilkan magnitude audio yang terlalu besar, melebihi batas kemampuan perangkat untuk mengeluarkan suara audio. Pada dasarnya, suatu perangkat akan menerjemahkan file audio menjadi tegangan tertentu yang ditentukan oleh magnitude audio tersebut. Akan tetapi, ketika magnitude audio terlalu besar, tegangan yang dihasilkan juga menjadi besar, melebihi batas tegangan pada perangkat. Alhasil, tegangan yang dihasilkan akan tersaturasi dan menyebabkan suara audio yang keluar dari perangkat menjadi terdistorsi atau pecah.

Kemudian, frekuensi passband yang dipilih adalah 50 Hz untuk f_{pass} dan 125 Hz untuk f_{stop} . Angka ini dipilih karena bass memiliki frekuensi antara 60-250 Hz [3]. Terlihat dari spektrum frekuensi sinyal output FIR, bahwa magnitude pada frekuensi 50 Hz naik sekitar 3 dB dari magnitude aslinya. Kemudian, terlihat juga bahwa sinyal dengan frekuensi di atas 125 Hz (perhatikan frekuensi 1000 Hz) memiliki magnitude yang lebih kecil dari sinyal aslinya. Bahkan, sinyal dengan frekuensi di atas 5000 Hz seakan-akan tertahan sempurna. Hal ini menunjukkan bahwa filter telah bekerja dengan baik. Berbeda dengan data, audio yang dihasilkan memiliki suara bass yang tidak terlalu kencang dan mencolok dibanding dengan suara musik lainnya. Hal ini disebabkan karena filter FIR tidak memiliki slope filter yang cukup tajam. Suara musik lain selain bass yang memiliki frekuensi di atas 250 Hz masih memiliki magnitude yang cukup besar, sehingga masih terdengar jelas dan agak menutupi suara bass.

B. Filter IIR MATLAB

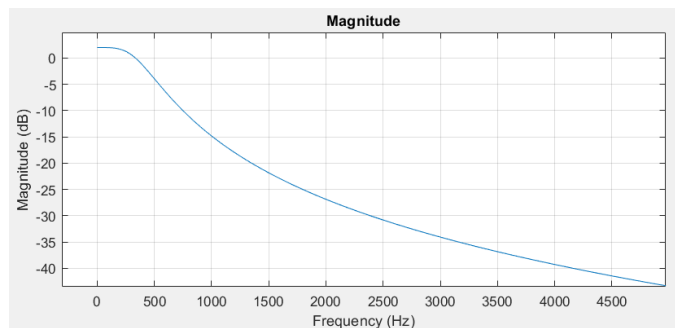


Figure III-4 Respons Frekuensi Filter IIR MATLAB

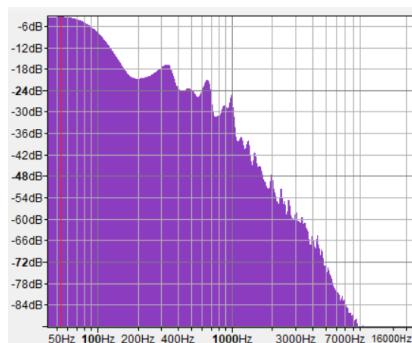


Figure III-5 Spektrum Frekuensi Audio Output Filter IIR MATLAB

```
iir_a_matlab = 1,-1.923,0.92586
iir_b_matlab = 0.00089908,0.0017982,0.00089908
```

Filter IIR MATLAB dibuat menggunakan filter jenis butterworth. Filter butterworth dipilih karena filter ini memiliki respons frekuensi paling halus dan ripple paling minim di antara filter IIR lainnya [4]. Hal ini terlihat dari respons frekuensi filter ini yang memiliki respons frekuensi yang halus serta tidak ada ripple.

Frekuensi yang dipilih untuk f_{pass} dan f_{stop} pada filter ini adalah masing-masing 250 Hz dan 750 Hz dengan atenuasi minimum nya adalah 12 dB. Nilai f_{pass} dipilih mendekati batas atas frekuensi bass karena filter IIR memiliki slope respons frekuensi yang besar. Akibatnya, sinyal dengan frekuensi yang

lebih besar dari 250 Hz yaitu suara vokal penyanyi dan pengiring musik lainnya akan langsung dilemahkan sehingga suara bass terdengar jelas. nilai f_{stop} dipilih bernilai 750 Hz karena frekuensi tersebut berada di sekitar frekuensi suara vokal yang ingin dilemahkan. Suara vokal yang lemah akan memperkuat suara bass serta musik pengiring lainnya. Pelemahan sinyal pada filter dapat terlihat pada spektrum frekuensi sinyal output IIR MATLAB, yaitu pada frekuensi 750 Hz, magnitude sinyal diperkecil dari -26 dB menjadi -32 dB. Sinyal-sinyal lain dengan frekuensi di atas 750 Hz, juga diperlemah sehingga magnitude nya lebih kecil dibanding sinyal aslinya.

Filter ini memiliki orde yang ditentukan oleh fungsi buttord. Fungsi buttord menerima parameter frekuensi f_{pass} , f_{stop} , serta atenuasi minimum pada f_{stop} . Hasil dari fungsi buttord pada filter ini adalah filter yang bernilai dua serta frekuensi ternormalisasi pada filter. Walaupun memiliki orde yang rendah, koefisien filter yang dihasilkan dari filter IIR dengan orde ini sudah mampu melakukan filtering dengan baik, terlihat dari spektrum frekuensi sinyal output dan suara output yang dihasilkan filter ini. Hal ini disebabkan karena filter IIR lebih efisien secara komputasi sehingga meskipun jumlah koefisien filter nya lebih kecil dari IIR, kinerja keduanya bisa dibilang sama atau bahkan lebih baik. Selain itu, filter IIR juga menggunakan sinyal feedback dari output untuk diolah kembali bersama sinyal input, sehingga untuk mencapai respons frekuensi yang sama, filter IIR dapat menggunakan koefisien filter yang lebih sedikit dibanding filter FIR.

Filter IIR secara default memiliki penguatan sebesar nol dB untuk sinyal dengan frekuensi di bawah f_{pass} . Artinya, sinyal tersebut hanya akan diteruskan dari sinyal aslinya dan tidak diperbesar sama sekali. Oleh karena itu, agar suara bass terdengar lebih keras dari suara aslinya, perlu diberikan penguatan tambahan yang dilakukan dengan cara mengalikan koefisien filter IIR bagian numerator (b) agar terjadi penguatan untuk keseluruhan sinyal audio. Penguatan yang dipilih adalah 2 dB karena setelah dilakukan percobaan, penguatan yang lebih besar dari nilai tersebut akan menghasilkan sinyal audio yang pecah.

Perbedaan lain dari hasil pemfilteran IIR dengan filter FIR terlihat pada spektrum frekuensi pada frekuensi tinggi. Pada filter FIR, sinyal dengan frekuensi 5000 Hz ke atas sudah tertahan secara sempurna. Sedangkan filter IIR baru menahan sinyal pada sinyal dengan frekuensi 7000 Hz ke atas. Hal ini menunjukkan bahwa filter FIR memiliki slope respons frekuensi yang semakin besar seiring bertambahnya frekuensi sinyal sehingga plot respons frekuensi nya semakin menukik. Sebaliknya, filter IIR memiliki slope yang semakin kecil seiring bertambahnya frekuensi sinyal sehingga plot respons frekuensinya semakin melandai. Hal ini dapat memberikan pengaruh yang besar jika sinyal atau audio input yang ingin difilter memiliki frekuensi tinggi dengan magnitude yang besar. Pada penelitian ini, sinyal input berfrekuensi tinggi sudah memiliki magnitude yang kecil sehingga tidak terlalu berpengaruh pada sinyal output.

C. Filter IIR Manual

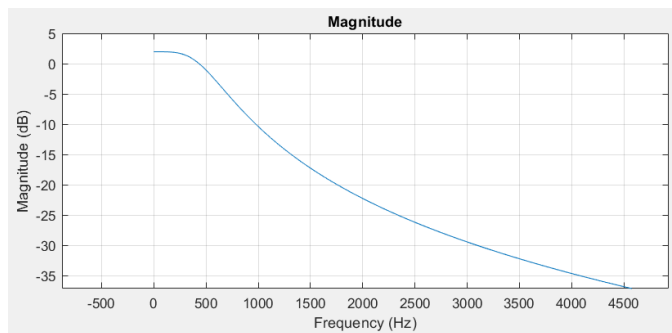


Figure III-6 Respons Frekuensi Filter IIR manual

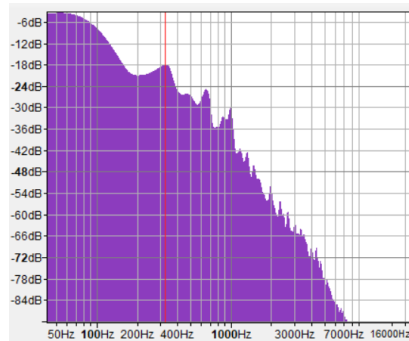


Figure III-7 Spektrum Frekuensi Audio Output Filter IIR manual

```
iir_a_manual = 1,-1.89935,0.904171
iir_b_manual = 0.001517,0.0030346,0.001517
```

Perhitungan koefisien filter IIR secara manual dimulai dengan terlebih dahulu menentukan spesifikasi filter. Pada penelitian ini, dipilih spesifikasi filter IIR yang sama dengan filter IIR MATLAB kemudian dilakukan perhitungan koefisien secara bilinear transform. Hasil yang didapatkan adalah koefisien filter IIR bagian denominator (a) sedikit lebih kecil dibanding koefisien dari MATLAB. Sedangkan koefisien filter IIR bagian numerator (b) cukup jauh lebih besar dibanding koefisien dari MATLAB. Galat yang dihasilkan koefisien a mencapai 2,34% sedangkan untuk koefisien b mencapai 68,72%. Hal ini disebabkan karena pada perhitungan koefisien, banyak terjadi pembulatan angka yang menyebabkan nilai koefisien tidak sesuai dengan nilai yang seharusnya. Selain itu, ukuran step pada integrasi numerical trapezoid (T) [5] yang tidak diketahui nilainya dan diasumsikan bernilai satu dapat menjadi sumber galat pada perhitungan koefisien.

Hasil simulasi MATLAB dengan koefisien filter dari perhitungan tangan menunjukkan bahwa filter tetap bekerja sebagaimana mestinya, yaitu sebagai low pass filter. Hasil filtering filter ini menunjukkan bahwa pelemahan mulai terjadi pada frekuensi di atas frekuensi f_{pass} yaitu 250 Hz. Perbedaan antara sinyal output hasil filtering IIR perhitungan tangan dengan MATLAB terletak pada magnitude sinyal yaitu sinyal dengan filter IIR perhitungan tangan magnitude nya lebih besar dibandingkan filter IIR MATLAB. Hal ini terjadi karena

koefisien IIR b pada perhitungan tangan lebih besar, sehingga gain nya juga akan lebih besar. Selain itu, slope pada filter IIR perhitungan tangan tidak sebesar filter IIR MATLAB. Terlihat dari plot respons frekuensi bahwa untuk frekuensi yang sama pelemahan filter IIR MATLAB lebih besar dibanding perhitungan tangan. Akibatnya, redaman suara musik dan vokal pada audio output tidak berjalan sebaik filter IIR MATLAB. Audio output filter IIR perhitungan tangan cukup pecah bass nya karena magnitude nya yang terlalu besar dan vokal serta musik pengiring masih cukup terdengar dan tidak teredam dengan baik. Filter IIR hasil perhitungan tangan bisa dikatakan tidak sesuai dengan spesifikasi karena gagal membuat audio yang ter-bassboosted dengan baik.

Tingginya galat dan kurangnya kualitas sinyal output filter IIR perhitungan tangan dapat diminimalisasi dengan cara menyimpan tiap variabel dan hasil hitungan pada suatu program dan menggunakan kalkulator yang lebih akurat seperti Excel, python, Wolphram atau MATLAB. Dengan menyimpan hasil setiap langkah perhitungan ke variabel, pembulatan angka akan sangat minim dan hasil perhitungan akan lebih akurat. Selain itu, bisa juga dilakukan perhitungan koefisien filter dengan metode lain yang memiliki perhitungan lebih akurat.

IV. SIMPULAN

Filter bass booster dapat dibuat dengan filter FIR maupun IIR dengan konfigurasi lowpass. Filter ini meredam sinyal dengan frekuensi yang lebih besar dari frekuensi bass. Selain itu, dilakukan juga penguatan pada seluruh sinyal dengan cara mengalikan koefisien filter dengan koefisien tertentu. Filter bass booster bekerja lebih baik dibanding IIR karena memiliki slope respons frekuensi yang lebih besar sehingga sinyal selain bass dapat diredam lebih baik dan suara bass akan lebih terdengar. Filter IIR juga lebih efisien secara komputasi karena menggunakan feedback. Filter IIR dapat dibuat menggunakan MATLAB maupun perhitungan. Akan tetapi, filter hasil perhitungan tangan melakukan filtering kurang baik karena banyaknya galat akibat pembulatan dan asumsi.

REFERENSI

- [1] Hutabarat, Mervin T., Petunjuk Praktikum EL 3110 Pengolahan Sinyal Digital edisi 2023-2024, LDTE ITB, Bandung, 2023.
- [2] "Digital Signal Processing/Bilinear Transform." wikibooks.org. https://en.wikibooks.org/wiki/Digital_Signal_Processing/Bilinear_Transform (accessed November 22, 2023)
- [3] J. Smoot, "Understanding Audio Frequency Range in Audio Design." cuidevices.com. <https://www.cuidevices.com/blog/understanding-audio-frequency-range-in-audio-design> (accessed November 22, 2023)
- [4] P. Young, "Designing an Analogue Low Pass Filter System for a Quantum Ion Trap." sussex.ac.uk. <https://users.sussex.ac.uk/~pjly20/ras100.html> (accessed November 22, 2023)
- [5] R. Brown, "DSP : IIR Filter Desgin via Bilinear Transform." wpi.edu. https://spinlab.wpi.edu/courses/ece503_2014/10-3bilinear_transform.pdf (accessed November 22, 2023)

Lampiran

Perhitungan koefisien filter IIR dengan perhitungan tangan

$$f_{\text{sampling}} = 44100 \text{ Hz}$$

$$f_{\text{pass}} = 250 \text{ Hz}$$

$$f_{\text{stop}} = 750 \text{ Hz}$$

$$\text{Minimum Attenuation} = -12 \text{ dB} \rightarrow \delta = 0.2511886$$

$$\omega_p = \frac{2\pi f_{\text{pass}}}{f_s} = \frac{2\pi \cdot 250}{22050} = \frac{10\pi}{441} \text{ rad/s}$$

$$\omega_s = \frac{2\pi f_{\text{stop}}}{f_s} = \frac{2\pi \cdot 750}{22050} = \frac{30\pi}{441} \text{ rad/s}$$

Menghitung spek filter analog, $T \gg 1$

$$\Omega_p = \frac{2}{T} \tan \frac{\omega_p}{2} = 2 \tan \left(\frac{5\pi}{441} \right) = 0.07127$$

$$\Omega_s = \frac{2}{T} \tan \frac{\omega_s}{2} = 2 \tan \left(\frac{15\pi}{441} \right) = 0.214531$$

Mencari orde filter

$$N = \left\lceil \frac{\log \left(\frac{1}{\delta^2} - 1 \right)}{2 \log \left(\frac{\Omega_s}{\Omega_p} \right)} \right\rceil = \left\lceil \frac{\log \left(\left(\frac{1}{(0.214531)^2} - 1 \right) \right)}{2 \log \left(\frac{0.214531}{0.07127} \right)} \right\rceil = \lceil 1.22431 \rceil = 2$$

Mencari pole filter

$$s_k = \omega_p e^{j \left(\frac{\pi}{2} + \frac{(2k+1)\pi}{2N} \right)}$$

$$N = 2$$

$$k = 0 \text{ dan } 1$$

$$s_0 = \omega_p e^{j \left(\frac{\pi}{2} + \frac{\pi}{4} \right)}$$

$$= 0.07127 \left(\cos \frac{3\pi}{4} + j \sin \frac{3\pi}{4} \right)$$

$$= -0.050395 + j 0.050395$$

$$s_1 = \omega_p e^{j \left(\frac{\pi}{2} + \frac{3\pi}{4} \right)}$$

$$= 0.07127 \left(\cos \frac{5\pi}{4} + j \sin \frac{5\pi}{4} \right)$$

$$= -0.050395 - j 0.050395$$

$$H(s) = \frac{1}{(s - s_0)(s - s_1)}$$

$$= \frac{1}{(s + 0.050395 + j 0.050395)(s + 0.050395 - j 0.050395)}$$

$$= \frac{1}{s^2 + 0.100791s + (5.0794 \times 10^{-3})}$$

$$s = \frac{2}{T} \left(\frac{1-z^{-1}}{1+z^{-1}} \right), T=1$$

$$\begin{aligned}
 H(z) &= \frac{1}{\left(2 \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \right)^2 + 0.100791 \cdot 2 \left(\frac{1-z^{-1}}{1+z^{-1}} \right) + (5.0794 \times 10^{-3})} \\
 &= \frac{1+z^{-1}+z^{-2}}{4(1-2z^{-1}+z^{-2}) + 0.201582(1-z^{-2}) + (5.0794 \times 10^{-3})(1+2z^{-1}+z^{-2})} \\
 &= \frac{1+z^{-1}+z^{-2}}{4.206614 - 7.9898142z^{-1} + 3.8034974z^{-2}}
 \end{aligned}$$

$$\text{Gain} = 2\text{dB} \rightarrow 1.258925$$

$$1.258925 = K \frac{1+z^{-1}+z^{-2}}{4.206614 (1 - 1.89935z^{-1} + 0.904171z^{-2})}$$

$$1.258925 = k \left(\frac{1+z^{-1}+z^{-2}}{1 - 1.89935z^{-1} + 0.904171z^{-2}} \right) \Rightarrow k = 0.001517$$

$$\therefore H(z) = \frac{0.001517 + 0.0030346z^{-1} + 0.001517z^{-2}}{1 - 1.89935z^{-1} + 0.904171z^{-2}}$$

$$b = [0.001517 \quad 0.0030346 \quad 0.001517]$$

$$a = [1 \quad -1.89935 \quad 0.904171]$$