

Implementasi Metode Transformasi Bilinear Pada Filter Digital Infinite Impulse Response (IIR) Menggunakan Raspberry Pi

Murni Sri Rahayu Islamiyah¹, Rachmad Saptono², Hadiwiayatno³

^{1,2} Program Studi Jaringan Telekomunikasi Digital,

Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

³ Program Studi Teknik Telekomunikasi,

Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

Murnirahayu120@gmail.com, rachmad.saptono@polinema.ac.id, hadiwiayatno@polinema.ac.id

Abstract— The process of sending information there is always interference or noise, the nature of noise cannot be eliminated but can be minimized. In this study, the author uses an IIR Digital Filter because the IIR filter has an impulse of infinite duration so that it can be matched with an analog filter which also has an impulse response of infinite duration. 1 and chebyshev type 2 on MATLAB. MATLAB code is used to create a filter design program according to the specifications that have been made. MATLAB Simulink is used to create a block filter design to be implemented on the Raspberry Pi. Raspberry pi is used as a device to process the filter. The results of running MATLAB code and also Simulink are compared to see if the implementation results are the same as the program results. The types of filters that are good for use in IIR digital filters are Butterworth filters and Chebyshev 2 filters because there is no ripple in the passband. The type of filter that is not good for use in digital IIR filters is the Chebyshev 1 filter. The effect of ripple changes on the order value is that the smaller the ripple value, the larger the resulting order value. The effect of changing the stopband attenuation is that the more the stopband attenuation, the smaller the resulting order.

Keywords— IIR, Transformasi Bilinear, Butterworth, Chebyshev1, Chebyshev2, Raspberry Pi

Abstrak — Proses pengiriman informasi selalu terdapat gangguan atau noise, sifat noise tidak dapat dihilangkan namun dapat diminimalisir. Dalam penelitian ini penulis menggunakan Filter Digital IIR dikarenakan filter IIR memiliki impulse berdurasi tak hingga sehingga dapat dicocokkan dengan filter analog yang juga memiliki respon impuls durasi tak hingga. Pada penelitian ini akan dirancang Filter Digital IIR menggunakan metode transformasi bilinear dengan respon filter butterworth, chebyshev tipe 1 dan chebyshev tipe 2 pada MATLAB. MATLAB code digunakan untuk membuat program design filter sesuai dengan spesifikasi yang telah dibuat. MATLAB Simulink digunakan untuk membuat design block filter untuk di implementasikan pada Raspberry Pi. Raspberry pi digunakan sebagai perangkat untuk memproses filter tersebut. Hasil dari running MATLAB code dan juga Simulink dibandingkan untuk melihat apakah hasil implementasi sama dengan hasil program. Dari hasil pengujian dapat diketahui hasil respon frekuensi pada berbagai jenis filter yang digunakan pada pengujian sesuai dengan karakteristik masing-masing filter. Jenis filter yang baik digunakan pada filter digital IIR adalah filter Butterworth dan filter Chebyshev 2 karena pada passband tidak terdapat ripple. Jenis filter yang tidak baik digunakan pada filter digital IIR adalah filter Chebyshev 1. Pengaruh perubahan ripple terhadap nilai orde yaitu semakin kecil nilai ripple semakin besar nilai orde yang dihasilkan. Pengaruh perubahan attenuasi stopband yaitu semakin attenuasi stopband semakin kecil orde yang dihasilkan.

Kata kunci— IIR, Bilinear Transform, Butterworth, Chebyshev1, Chebyshev2, Raspberry Pi

I. PENDAHULUAN

Perkembangan teknologi saat ini sangat pesat contohnya pada bidang komunikasi, sekarang manusia bisa dengan mudah untuk berkomunikasi secara jarak jauh. Pada komunikasi, sinyal memegang peranan penting, yaitu sebagai pembawa informasi. Pada proses pengiriman informasi selalu terdapat gangguan atau noise, sifat noise tidak dapat dihilangkan namun dapat diminimalisir.[1] Noise dapat mengakibatkan informasi yang diterima tidak sesuai dengan yang dikirimkan. Agar sinyal dapat sesuai kebutuhan dan keinginan maka memerlukan sebuah filter untuk memilah sinyal dari noise. Filter merupakan sebuah rancangan untuk meloloskan atau menyaring sinyal masukan agar sinyal yang masuk sesuai dengan frekuensi yang diinginkan.[2] Filter dibagi menjadi 2 yaitu filter analog dan filter digital.

Filter digital memiliki kelebihan yaitu lebih mudah diubah dan programmable, filter digital juga menggunakan memori processor jadi lebih mudah di desain, tes dan

diimplementasikan atau dihubungkan pada komputer. Filter digital lebih stabil, relatif lebih simple dan compact.[1] Filter Digital memiliki 2 tipe, salah satunya yaitu IIR (Infinite Impulse Response). Kelebihan Filter Digital IIR yaitu memiliki respon impuls tak berhingga, sehingga dapat dicocokkan dengan filter analog yang juga memiliki respon impuls durasi tak berhingga. Filter IIR juga membutuhkan koefisien yang lebih sedikit untuk respon frekuensi yang curam sehingga dapat mengurasi jumlah waktu komputasi selain itu implementasi filter IIR membutuhkan lebih sedikit parameter, sehingga lebih sedikit memori yang digunakan.

Dalam perancangan sebuah filter dapat digunakan simulasi perancangan melalui perangkat lunak yang bertujuan untuk mempermudah dalam penentuan desain rangkaian yang akan digunakan untuk merangkai sebuah filter dan dapat menjelaskan karakteristik dan respon dari suatu filter.[3] Salah satu perangkat lunak yang dapat

melakukan simulasi perancangan tersebut adalah MATLAB.

Pada penelitian ini akan dirancang Filter Digital IIR menggunakan metode transformasi bilinear dengan respon filter butterworth, chebyshev tipe 1 dan chebyshev tipe 2 pada software MATLAB kemudian filter akan diimplementasikan pada Raspberry Pi. Raspberry Pi memiliki kelebihan dibanding dengan DSP Kit yang lain yakni mampu melakukan komputasi yang lebih cepat, memori yang lebih besar, dan bahasa pemrograman yang lebih mudah dipahami.

Hasil perancangan dan implementasi pada penelitian ini, nantinya dapat mengetahui perbandingan respon frekuensi dari software MATLAB dengan respon frekuensi dari Raspberry Pi atau sinyal yang sudah mengalami pemfilteran, juga perbandingan bentuk respon frekuensi sebelum dan sesudah mengalami pemfilteran.

II. TINJAUAN PUSTAKA

A. Filter

Filter adalah sebuah system atau jaringan yang secara selektif merubah karakteristik (bentuk gelombang frekuensi, fase dan amplitudo) dari sebuah sinyal. Berdasarkan frekuensi yang dilewatkan, secara umum Filter terbagi menjadi :

- 1) *Low Pass Filter* atau sering disingkat dengan LPF adalah Filter atau Penyaring yang melewatkan sinyal frekuensi rendah dan menghambat atau memblokir sinyal frekuensi tinggi.
- 2) *Band Pass Filter* atau sering disingkat dengan BPF adalah Filter atau penyaring frekuensi yang melewatkan sinyal frekuensi dalam rentang frekuensi tertentu yaitu melewatkan sinyal yang berada diantara frekuensi batas bawah hingga frekuensi batas atasnya.

B. Filter Digital IIR

Filter digital adalah sebuah implementasi algoritma matematik ke dalam perangkat keras dan/atau perangkat lunak yang beroperasi pada sebuah input sinyal digital agar tujuan pemfilteran tercapai.[6] Salah satu contoh filter digital adalah Filter Infinite Impulse Response (IIR). Filter IIR dikarakteristikan oleh respon impuls berdurasi tak-terhingga. Beberapa respon impuls ini dapat dimodelkan dengan fungsi-fungsi sistem rasional atau dengan persamaan-persamaan perbedaan. Pada pemrosesan sinyal digital, filter-filter IIR mengimplikasikan operasi rekursif yang dapat di implementasikan secara efisien. Adapun beberapa respon filter IIR :

- 1) Butterworth adalah jenis Filter pemrosesan sinyal yang dirancang untuk memiliki respons frekuensi sedatar mungkin dalam passband. Ini juga disebut sebagai filter magnitudo rata maksimal.
- 2) Filter Chebyshev dapat dibagi menjadi dua macam, yaitu filter Chebyshev tipe 1 yang mempunyai kerutan pada daerah frekuensi yang diloloskan (passband) dan filter Chebyshev tipe 2 yang mempunyai kerutan pada daerah yang diredam (stopband). Filter Chebyshev menghasilkan gain roll-off yang lebih tinggi di atas F_c .

Adapun salah satu metode pendekatan pada filter IIR adalah Metode Transformasi Bilinear. Transformasi bilinear digunakan dalam pemrosesan sinyal digital untuk mengubah representasi sistem waktu kontinu menjadi waktu diskrit dan sebaliknya. Transformasi bilinear juga memiliki efek non-linier, karena itu perlu dikompensasi sebelum dilakukan transformasi (pre-wrapping). Prewrapping digunakan sebagai kompensasi terhadap efek non linier dari transformasi bilinear, yaitu dengan prewrapping frekuensi sebelum dilakukan transformasi.

C. Noise atau Derau

Noise (derau) merupakan sinyal lain yang tidak diharapkan dalam sistem telekomunikasi karena bersifat mengganggu terhadap sinyal asli serta kehadirannya tidak bisa ditentukan (acak). Banyaknya noise tidak dapat ditentukan secara pasti, hanya dapat dirumuskan probabilitas ataupun kisaran nilai (range) nya saja.

Gangguan yang diakibatkan oleh noise dapat mengubah sinyal informasi, yang menyebabkan gelombang sinus mempunyai sinyal derau yang kecil yang bergabung didalamnya. Sehingga penerima tidak dapat membedakan sinyal informasi yang sebenarnya dari derau yang ditambahkan.

Salah satu contoh noise pada audio adalah Additive White Gaussian Noise. Additive White Gaussian Noise (AWGN) merupakan noise yang pasti terjadi dalam jaringan nirkabel manapun, memiliki sifat-sifat Additive, White, dan Gaussian. Sifat Additive artinya noise ini dijumlahkan dengan sinyal, sifat White artinya noise tidak bergantung pada frekuensi sistem operasi dan memiliki rapat daya yang konstan, dan sifat Gaussian artinya besarnya tegangan noise memiliki rapat peluang terdistribusi gaussian.

D. Raspberry Pi

Raspberry Pi (Raspi) adalah komputer papan tunggal (single-board circuit; SBC) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi menggunakan sytem on a chip (SoC) dari Broadcom BCM2835 hingga BCM 2837 (Raspberry Pi 3), juga sudah termasuk prosesor ARM1176JZF-S MHz bahkan 1.2GHz 64-bit quad-core ARMv8 CPU untuk Raspberry Pi 3, GPU VideoCore IV dan kapasitas RAM hingga 1 GB.

E. Python

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Cara mudah yang bisa digunakan untuk membuat script sebelum belajar bahasa Pemrograman Python supaya bisa lebih paham dan mengerti dengan segera tanpa menghabiskan waktu yang lama [6].

F. MiFi

MiFi adalah sebuah modem *mobile*, di mana berfungsi untuk menghasilkan radiasi hotspot yang bersifat portabel dan menggunakan koneksi data seluler. Dalam kata lain, MiFi bisa disebut sebagai WiFi portabel yang bisa dibawa kemana-mana. Istilah MiFi sendiri adalah sebuah singkatan dari “Mobile WiFi”.

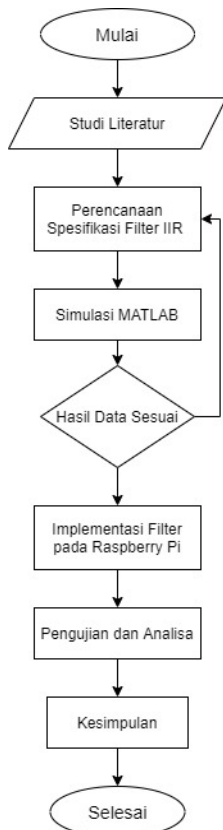
G. Kabel LAN

Kabel LAN (local area network) atau yang biasa kita sebut dengan kabel utp adalah kabel yang berguna untuk menghubungkan beberapa komputer/perangkat dalam area yang terbatas contohnya seperti rumah, kantor, laboratorium, perkantoran.kabel lan paling sering digunakan membuat sebuah jaringan komputer yang bersetandar pada topologi jaringan.

III. METODE PENELITIAN

A. Rancangan Penelitian

Tahapan penelitian yang akan dilakukan dinyatakan dengan diagram alir yang tertera pada Gambar 1.



Gambar 1 Flowchart Rancangan Penelitian

Alur rancangan penelitian dijelaskan sebagai berikut :

- 1) Studi literature

Studi literatur yang diperlukan yaitu penulis memahami karakteristik dari filter digital IIR yang akan digunakan seperti orde, jenis dan respon filter.
- 2) Perencanaan Spesifikasi Filter

Perencanaan Spesifikasi Filter membahas mengenai spesifikasi filter digital IIR menggunakan metode

transformasi bilinear yang akan diimplementasikan pada Raspberry Pi.

- 3) Simulasi MATLAB

Pada tahap ini dilakukan simulasi filter pada MATLAB untuk membandingkan apakah hasil keluaran filter sesuai dengan yang diinginkan.
- 4) Implementasi filter

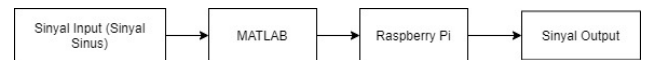
Pada tahap ini spesifikasi filter digital IIR menggunakan metode transformasi bilinear yang telah sesuai dengan yang di inginkan diimplementasikan pada perangkat Raspberry Pi.
- 5) Pengujian dan Analisa

Pada tahap ini dilakukan analisa hasil pengujian implementasi dengan membandingkan hasil keluaran pada MATLAB dan hasil keluaran pada Raspberry Pi.
- 6) Kesimpulan

Kesimpulan yang diambil berdasarkan pengujian yang dilakukan oleh penulis. Data grafik akan dibandingkan. Kemudian disimpulkan mengenai hasil yang telah diperoleh.

B. Rancangan Sistem

Rancangan sistem membahas sistem yang akan digunakan pada penelitian ini, rancangan sitem dapat dilihat pada gambar 2.



Gambar 2 Blok Diagram Sistem

Rancangan sistem pada penelitian ini :

- 1) Menginputkan spesifikasi filter

Sinyal Sinus sebagai sinyal input pada penelitian ini menggunakan frekuensi sampling 48KHz, pada lowpass filter menggunakan frekuensi cut off 4000 Hz hingga 9000 Hz dan pada bandpass filter menggunakan frekuensi cut off 10KHz hingga 15KHz dengan berbagai ripple dan Atenuasi stopband untuk melihat pengaruhnya terhadap respon frekuensi yang dihasilkan.

```

FT = input ('Masukkan frekuensi sampling = ');
Fp = input ('Masukkan frekuensi batas passband (Hz) = ');
Fs = input ('Masukkan frekuensi batas stopband (Hz) = ');
Rp = input ('Masukkan ripple passband (dB) = ');
Rs = input ('Masukkan atenuasi stopband (dB) = ');
  
```

Orde filter yang didapatkan dari berbagai spesifikasi sinyal yang di inputkan dan jenis respon filter yang digunakan.

```

% [N, Wn] = buttord(Wp, Ws, Rp, Rs, 's');
% [N, Wn] = cheb1ord(Wp, Ws, Rp, Rs, 's');
[N, Wn] = cheb2ord(Wp, Ws, Rp, Rs, 's');
  
```

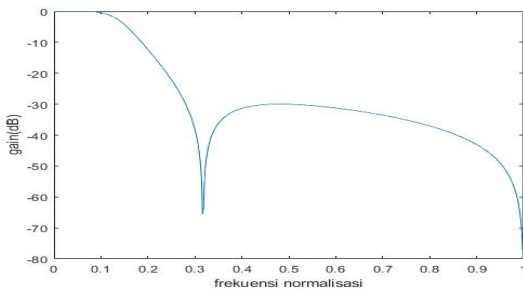
Melakukan pemfilteran dengan memilih jenis respon filter yang akan digunakan antara lain filter Butterworth, Chebyshev tipe 1, dan Chebyshev tipe 2 dengan mengcomment jenis filter yang tidak digunakan.

```
%[b,a] = butter(N,Wp,'low');
%[b,a]=cheby1(N,Rp,Wn,'low');
[b,a]=cheby2(N,Rs,Wn,'low');
```

Melakukan pemfilteran dengan jenis filter yang telah dipilih menggunakan metode Transformasi Bilinier

```
[B,A] = freqz(b,a,512);
Hasil display running program MATLAB
disp('Orde filter = '); disp(N)
disp('Koefisien pembilang = '); disp(b)
disp('Koefisien penyebut = '); disp(a)
plot(A/pi,mag)
xlabel('frekuensi normalisasi')
ylabel('gain(dB)')
```

Contoh hasil keluaran simulasi MATLAB ditampilkan pada gambar 3.



Gambar 3 contoh hasil simulasi MATLAB

Dihasilkan koefisien-koefisien filter setelah program MATLAB di running.

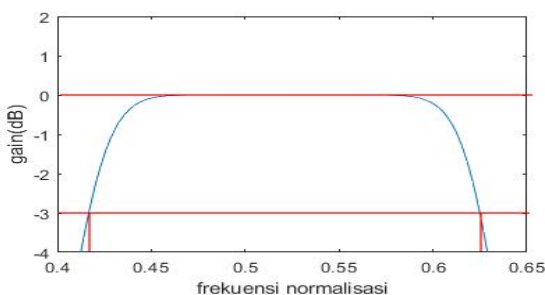
```
Koefisien pembilang =
    0.0406    -0.0242     0.0406

Koefisien penyebut =
    1.0000    -1.6393     0.6964
```

Gambar 3.4 Koefisien-Koefisien filter yang dihasilkan

IV. HASIL PENGUJIAN

A. Hasil pengujian pada simulasi matlab

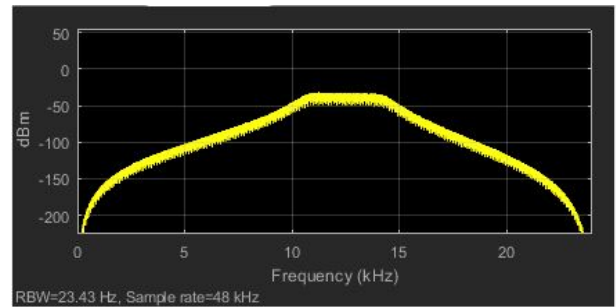


Gambar 4. BPF dengan jenis respon filter butterworth M-File

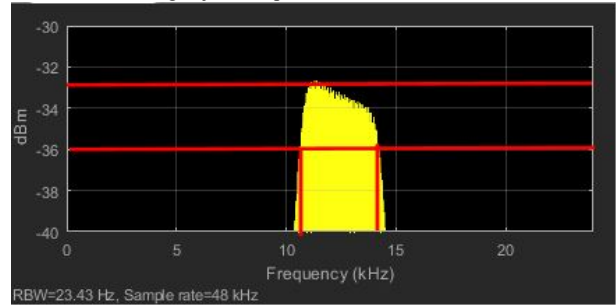
Frekuensi normalisasi = $2 \times F_c / F_s$

$F_c 1 : 0.42 = 2 \times F_c / 48000 = 10080 \text{ Hz}$

$F_c 2 : 0.625 = 2 \times F_c / 48000 = 15000 \text{ Hz}$

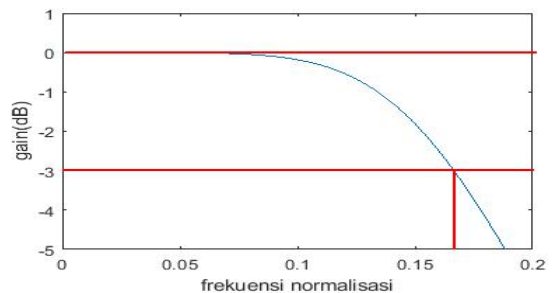


Gambar 5. BPF dengan jenis respon filter butterworth Simulink



Gambar 6. BPF dengan jenis respon filter butterworth -3dB Simulink

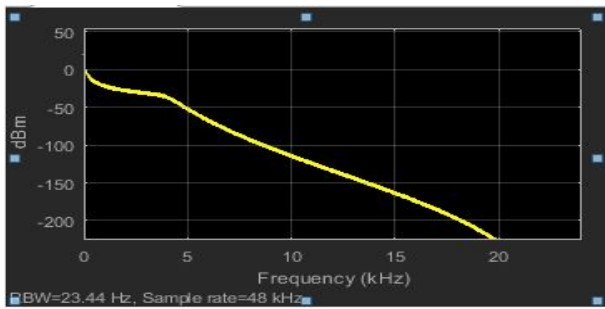
Grafik diatas adalah hasil output sinyal BPF dengan ripple 15 dB dan attenuasi stopband 30 dB dapat terlihat frekuensi terfilter dengan baik dengan tidak terdapat ripple pada sebelum dan sesudah filter. Untuk mengetahui nilai frekuensi cut off dengan menarik garis lurussumbu y pada puncak sinyal kemudian dikurangi minus 3 dB dan ditarik daris lurus pada sumbu y. Dari grafik MATLAB code dapat diketahui nilai frekuensi cut off 1 adalah 10080 Hz dan frekuensi cut off 2 adalah 15000 Hz, kemudian diperoleh bandwidth dengan mengurangi nilai $f_c 2$ dikurangi $f_c 1$ yakni 4920 Hz. Dari grafik Simulink diperoleh nilai $F_c 1$ 11050 Hz dan nilai $F_c 2$ adalah 14050 Hz, sehingga diperoleh nilai bandwidth adalah 4000 Hz. Frekuensi tersebut masuk ke dalam frekuensi suara. Dari hasil grafik juga dapat diketahui bahwa filter butterworth adalah filter yang baik digunakan untuk filter digital.



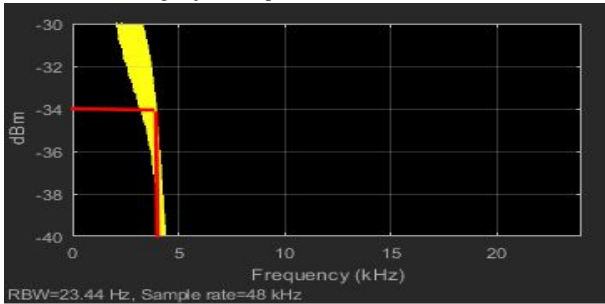
Gambar 7. LPF dengan jenis respon filter butterworth M-File

Frekuensi normalisasi = $2 \times F_c / F_s$

$F_c : 0.165 = 2 \times F_c / 48000 = 3690 \text{ Hz}$

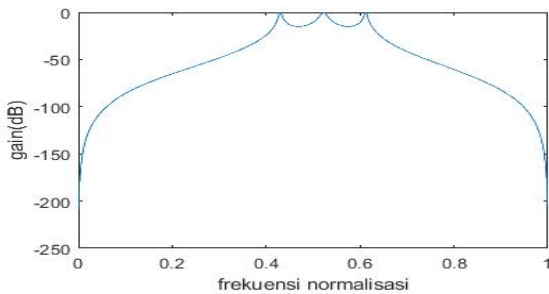


Gambar 8. LPF dengan jenis respon filter butterworth Simulink

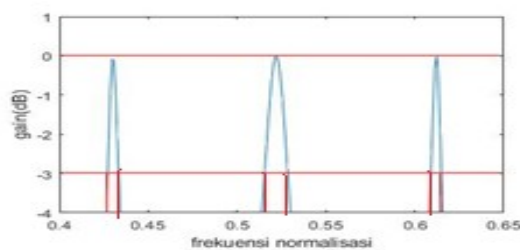


Gambar 9. LPF dengan jenis respon filter butterworth -3dB Simulink

Grafik diatas adalah hasil output sinyal LPF dengan ripple 15 dB dan attenuasi stopband 30 dB dapat terlihat frekuensi terfilter dengan baik dengan tidak terdapat ripple pada sebelum dan sesudah filter. Bentuk respon filter juga sesuai dengan spesifikasi lowpass filter. Dari grafik MATLAB code dapat diketahui nilai frekuensi cut off adalah 3690 Hz. Dari grafik Simulink diperoleh nilai F_c 4000 Hz. Frekuensi tersebut masuk ke dalam frekuensi suara. Dari hasil grafik juga dapat diketahui bahwa filter LPF butterworth adalah filter yang baik digunakan untuk filter digital.



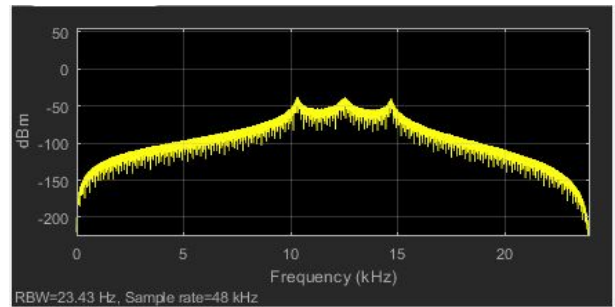
Gambar 10. BPF dengan jenis respon filter Chebyshev1 M-File



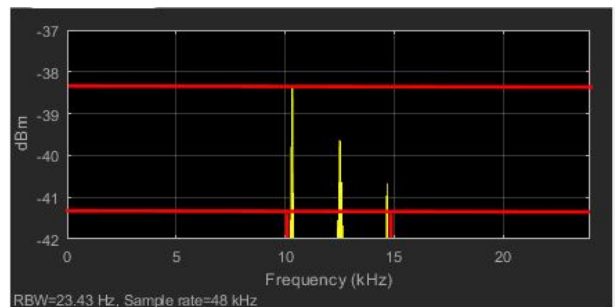
Gambar 11. BPF dengan jenis respon filter Chebyshev1 -3dB M-File

Frekuensi normalisasi = $2 \times F_c / F_s$
 Ripple 1: $(0.425 = 2 \times F_c / 48000 = 10200 \text{ Hz}) - (0.430 = 2 \times F_c / 48000 = 10320 \text{ Hz}) = 120 \text{ Hz}$

Ripple 2: $(0.515 = 2 \times F_c / 48000 = 12360 \text{ Hz}) - (0.53 = 2 \times F_c / 48000 = 12720 \text{ Hz}) = 360 \text{ Hz}$
 Ripple 3 : $(0.61 = 2 \times F_c / 48000 = 14640 \text{ Hz}) - (0.615 = 2 \times F_c / 48000 = 14760 \text{ Hz}) = 120 \text{ Hz}$

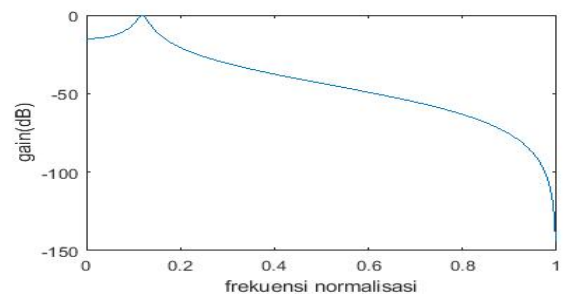


Gambar 12. BPF dengan jenis respon filter Chebyshev1 Simulink

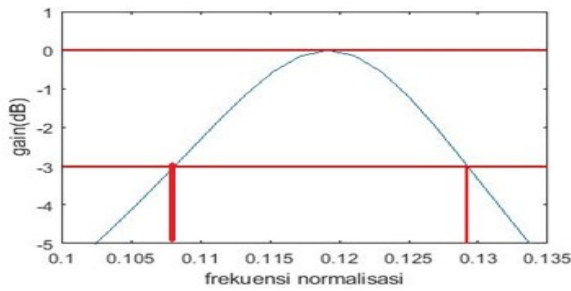


Gambar 13. BPF dengan jenis respon filter Chebyshev1 -3dB Simulink

Grafik diatas adalah hasil output sinyal BPF Chebyshev1 dengan ripple 15 dB dan attenuasi stopband 30 dB dapat terlihat frekuensi terfilter dengan baik dengan terdapat ripple frekuensi passband sesuai dengan karakteristik filter Chebyshev tipe 1. Dari grafik MATLAB code dapat diketahui terdapat 3 ripple yang berada pada titik -3dB. Hasil bandwidth pada ripple 1 adalah nilai frekuensi cut off 1, 10200 Hz dan frekuensi cut off 2, 10320 Hz. kemudian nilai $f_c 2$ dikurangi $f_c 1$ yakni 120 Hz. Bandwidth pada ripple 2 adalah nilai frekuensi cut off 1, 12360 Hz dan frekuensi cut off 2, 12720 Hz. kemudian nilai $f_c 2$ dikurangi $f_c 1$ yakni 360 Hz. Bandwidth pada ripple 3 adalah nilai frekuensi cut off 1, 14640 Hz dan frekuensi cut off 2, 14760 Hz. kemudian nilai $f_c 2$ dikurangi $f_c 1$ yakni 120 Hz. Dari grafik Simulink diperoleh 3 titik ripple pada -3 dB yakni pada frekuensi 10000 Hz, 13000 Hz, dan 15000 Hz. Frekuensi tersebut masuk ke dalam frekuensi suara. Dari hasil grafik juga dapat diketahui bahwa filter Chebyshev tipe 1 tidak baik digunakan untuk filter digital.



Gambar 14. LPF dengan jenis respon filter Chebyshev1 M-File

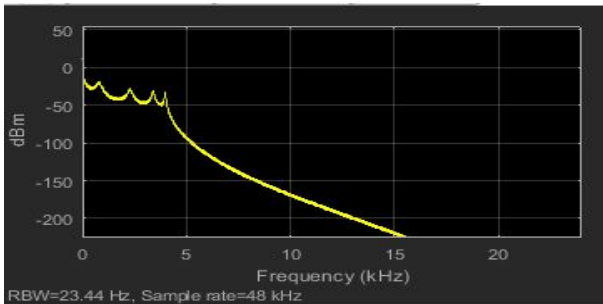


Gambar 15. LPF dengan jenis respon filter Chebyshev1 -3dB M-File

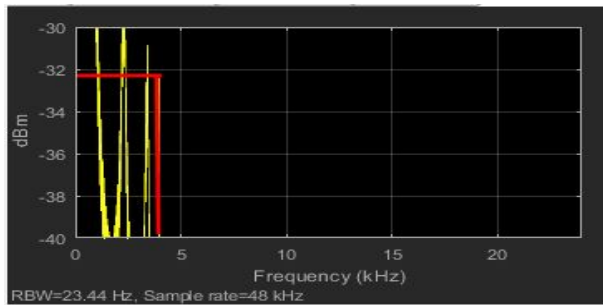
Frekuensi normalisasi = $2 \times F_c / F_s$

$F_c 1 : 0.108 = 2 \times F_c / 48000 = 2592 \text{ Hz}$

$F_c 2 : 0.129 = 2 \times F_c / 48000 = 3096 \text{ Hz}$

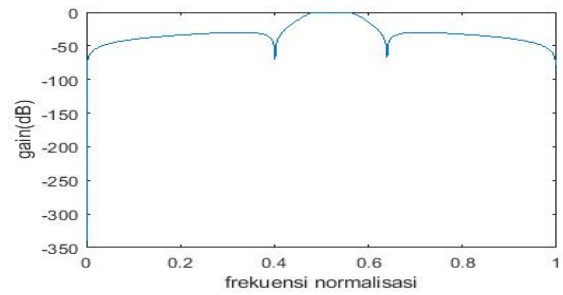


Gambar 16. LPF dengan jenis respon filter Chebyshev1 Simulink

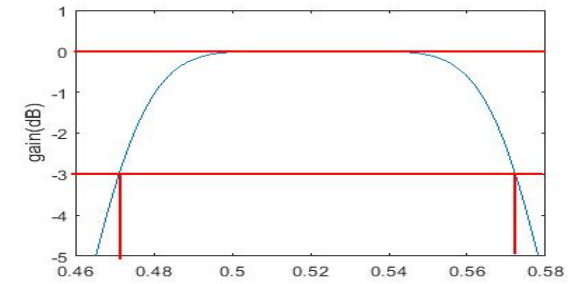


Gambar 17. LPF dengan jenis respon filter Chebyshev1 -3dB Simulink

Grafik diatas adalah hasil output sinyal LPF Chebyshev1 dengan ripple 15 dB dan attenuasi stopband 30 dB dapat terlihat frekuensi terfilter dengan baik dengan terdapat ripple pada frekuensi passband sinyal sesuai dengan karakteristik filter Chebyshev1. Pada grafik MATLAB code dapat diketahui nilai frekuensi cut off 1 adalah 2592 Hz dan frekuensi cut off 2 adalah 3096 Hz, kemudian diperoleh bandwidth dengan mengurangi nilai $f_c 2$ dikurangi $f_c 1$ yakni 504 Hz. Dari grafik Simulink terdapat 3 titik ripple pada frekuensi 2000 H, 3000 Hz, dan 4000 Hz. Frekuensi tersebut masuk ke dalam frekuensi suara. Dari hasil grafik juga dapat diketahui bahwa LPF chebyshev1 adalah filter yang tidak baik digunakan untuk filter digital.



Gambar 18. BPF dengan jenis respon filter Chebyshev2 M-File

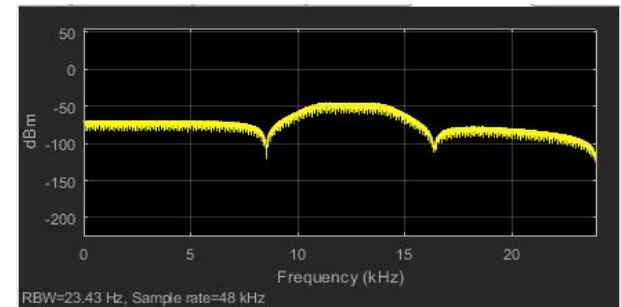


Gambar 19. BPF dengan jenis respon filter Chebyshev2 -3dB M-File

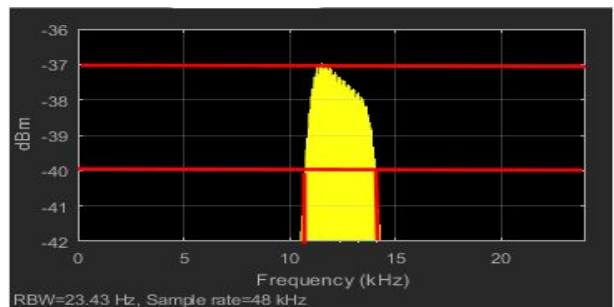
Frekuensi normalisasi = $2 \times F_c / F_s$

$F_c 1 : 0.47 = 2 \times F_c / 48000 = 11280 \text{ Hz}$

$F_c 2 : 0.57 = 2 \times F_c / 48000 = 13680 \text{ Hz}$



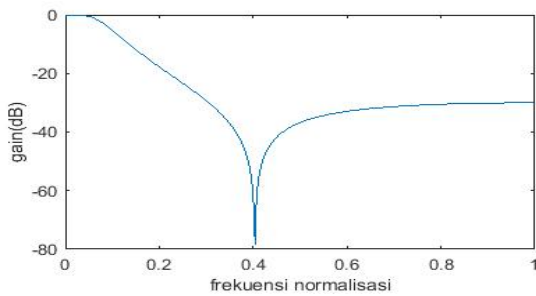
Gambar 20. BPF dengan jenis respon filter Chebyshev2 Simulink



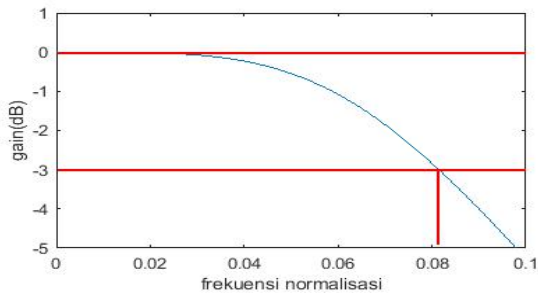
Gambar 21. BPF dengan jenis respon filter Chebyshev2 -3dB Simulink

Grafik diatas adalah hasil output sinyal BPF Chebyshev2 dengan ripple 15 dB dan attenuasi stopband 30 dB dapat terlihat frekuensi terfilter dengan baik dengan terdapat ripple pada sebelum dan sesudah filter. Hasil filter sinyal sesuai dengan karakteristik filter Chebyshev2. MATLAB code dapat diketahui nilai frekuensi cut off 1 adalah 11280 Hz dan frekuensi cut off 2 adalah 13680 Hz, kemudian diperoleh bandwidth dengan mengurangi nilai $f_c 2$ dikurangi $f_c 1$ yakni 2400 Hz. Dari grafik Simulink diperoleh nilai $F_c 1$ yakni 11000 Hz dan nilai $F_c 2$ adalah 14000 Hz, sehingga diperoleh nilai bandwidth adalah 4000 Hz. Frekuensi tersebut masuk ke dalam frekuensi suara. Dari

hasil grafik juga dapat diketahui bahwa BPF Chebyshev2 adalah filter yang baik digunakan untuk filter digital.



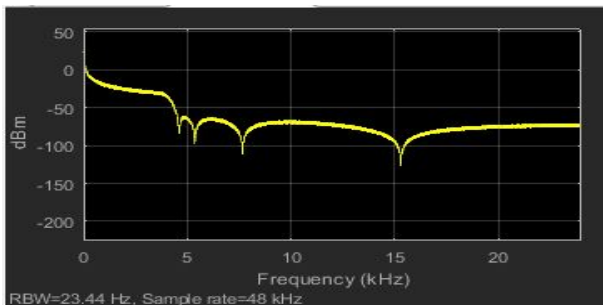
Gambar 22. LPF dengan jenis respon filter Chebyshev2 M-File



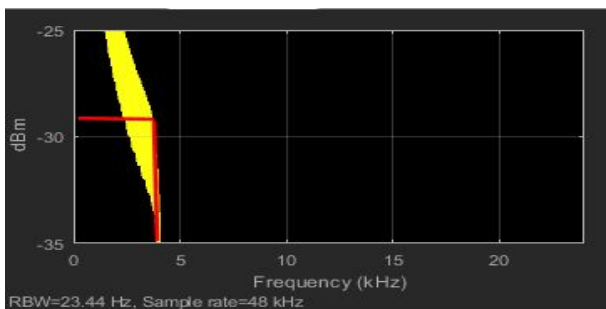
Gambar 21. LPF dengan jenis respon filter Chebyshev2 M-File

$$\text{Frekuensi normalisasi} = 2 \times F_c / F_s$$

$$F_c : 0.082 = 2 \times F_c / 48000 = 1968 \text{ Hz}$$



Gambar 22. LPF dengan jenis respon filter Chebyshev2 Simulink



Gambar 23. BPF dengan jenis respon filter Chebyshev2 -3dB Simulink

Grafik diatas adalah hasil output sinyal LPF dengan ripple 15 dB dan attenuasi stopband 30 dB dapat terlihat frekuensi terfilter dengan baik dengan terdapat ripple pada

sebelum dan sesudah filter sesuai dengan karakteristik Chebyshev2 filter. Dari grafik MATLAB code dapat diketahui nilai frekuensi cut off adalah 1968 Hz. Dari grafik Simulink diperoleh nilai F_c 4000 Hz. Frekuensi tersebut masuk ke dalam frekuensi suara. Dari hasil grafik juga dapat diketahui bahwa filter LPF Chebyshev2 adalah filter yang baik digunakan untuk filter digital

V. KESIMPULAN

Hasil respon frekuensi pada masing-masing jenis filter telah sesuai dengan karakteristik filter. Pada filter Butterworth frekuensi di luar frekuensi yang dilewatkan mengalami peredaman dan tidak terdapat ripple. Pada filter Chebyshev tipe 1 frekuensi di luar frekuensi cut off mengalami peredaman dan terdapat ripple pada frekuensi yang di lewatkan. Pada filter Chebyshev tipe 2 frekuensi di luar frekuensi yang dilewatkan mengalami peredaman dan terdapat ripple pada frekuensi yang di redam. Jenis filter yang baik digunakan pada filter digital Infinite Impulse Response adalah filter Butterworth dan filter Chebyshev2 karena pada passband tidak terdapat ripple. Jenis filter yang tidak baik digunakan pada filter digital IIR adalah filter Chebyshev 1. Lebar bandwidth berpengaruh pada hasil respon frekuensi, semakin sempit bandwidth hasil respon sinyal semakin buruk karena akan semakin banyak ripple yang dihasilkan.

REFERENSI

- [1] A. Bustamin and A. A. Prayogi, "Perbandingan Kinerja Filter Butterworth Berdasarkan Spesifikasi Frekuensi untuk Pengolahan Sinyal Suara," *Techno.Com*, vol. 18, no. 4, pp. 332–339, 2019.
- [2] A. A. Bimantara, M. S. Adhi, D. Priambodo, H. M. Azhar, and A. Junaidi, "Simulasi Penghilangan Noise Pada Sinyal Suara Menggunakan Metode Fast Fourier Transform (FFT)," *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 1, no. 2, pp. 20–25, 2019.
- [3] G. Ariyanto and H. Supriyono, "Implementasi Filter Digital Infinite Impulse Response pada DSP TMS320C6711," no. January 2016, pp. 33–38, 2002.
- [4] L. Lidyawati, A. R. Darlis, and A. F. Tamba, "Implementasi Filter Infinite Impulse Response (Iir) Dengan Respon Butterworth Dan Chebyshev Menggunakan Dsk Tms320C6713," *J. Elektro dan Telekomun. Terap.*, vol. 2, no. 1, pp. 95–103, 2016.
- [5] J. S. Komputer, F. I. Komputer, U. B. Nusantara, and J. Barat, "Adaptive IIR Filter Untuk Active Noise Controller Menggunakan Prosesor Sinyal Digital TMS320C542," no. 9.
- [6] A Hariyadi et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1073 012050.