

DOI: doi.org/10.58797/pilar.0101.02

Kajian Discrete Fourier Transform untuk Menganalisis Sinyal Arbitrer

Irsya Luthfiah Ramadhyagita*, Annisa Annisa, Faishal Kamindra,
Farhan Muhammad Rizky

Program Studi Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Jakarta

*Corresponding Email: irsyaluthfiah03@gmail.com

Received: 15 Februari 2022
Revised: 11 April 2022
Accepted: 23 Mei 2022
Online: 30 Juni 2022
Published: 30 Juni 2022

**Mitra Pilar: Jurnal
Pendidikan, Inovasi, dan
Terapan Teknologi**
p-ISSN: 2964-7622
e-ISSN: 2964-6014



Abstract

This study aims to examine the Discrete Fourier Transform using arbitrary signals. Signal processing is a mathematical operation performed on a signal to obtain the required information. In this case, a transformation occurs. The Fourier transform is a popular method often used to change the time form to the frequency form intended to facilitate analysis. The Fourier transform is divided into the Continuous Fourier and the Discrete Fourier. The program developed by the researcher, DFT, analyzes arbitrary signals created in two programs. The first program is a function of the DFT and the second is the main program used to generate DFT graphs. In this project, researchers will analyze arbitrary signals decomposed into sine waves. This program uses two main libraries, namely matplotlib, and NumPy. In this case, we have used the DFT method and plotted the results of the calculations performed. Now the researcher will analyze the process that the researcher did while using the DFT method and plot the results. Using DFT, arbitrary signals can be arranged into a series of sinusoids, each with a different frequency. The DFT method can arrange signals into a series of sinusoids with different frequencies. In the DFT amplitude spectrum, the signal frequency is shown as a vertical bar, with the height being the signal amplitude in the time domain. DFT can convert a sequence of equally spaced signals into information about the frequencies of all sine waves needed to sum the time domain signals.

Keywords: amplitude, arbitrary signals, Discrete Fourier Transform (DFT), frequency.

Abstrak

Penelitian ini bertujuan untuk mengkaji Discrete Fourier Transform dengan menggunakan sinyal arbitrer. Pengolahan sinyal adalah suatu operasi matematik yang dilakukan terhadap suatu sinyal sehingga diperoleh suatu informasi yang dibutuhkan. Dalam hal ini terjadi suatu transformasi. Transformasi Fourier merupakan salah satu metode populer yang sering digunakan untuk mengubah bentuk waktu bentuk frekuensi yang ditujukan untuk mempermudah analisis. Transformasi Fourier dibagi menjadi 2 bagian yaitu Fourier Kontinu dan Fourier Diskret. Program yang dikembangkan oleh peneliti yaitu, DFT untuk menganalisis sinyal arbitrer yang dibuat dalam dua program. Program pertama adalah fungsi dari DFT dan program kedua adalah program utama yang digunakan untuk memunculkan grafik DFT. Dalam project ini peneliti akan menganalisis sinyal arbitrer yang sudah diuraikan menjadi gelombang sinus. Program ini menggunakan dua library utama, yaitu matplotlib dan numpy. Pada case kali ini, telah menggunakan metode DFT dan membuat hasil plot dari perhitungan yang dilakukan. Sekarang peneliti akan menganalisis proses yang peneliti lakukan selama menggunakan metode DFT dan membuat hasil plot. Sebelum membahas hasil perhitungan, mari kembali membahas metode DFT itu sendiri. Dengan menggunakan DFT, dapat disusun sinyal arbitrer menjadi serangkaian sinusoid dan masing-masing akan memiliki frekuensi yang berbeda. Metode DFT dapat menyusun sinyal menjadi serangkaian sinusoid yang memiliki frekuensi yang berbeda-beda. Dalam spektrum amplitude DFT, frekuensi sinyal ditampilkan sebagai batang vertikal dengan ketinggiannya adalah amplitude sinyal dalam domain waktu. DFT dapat mengubah urutan sinyal dengan jarak yang sama menjadi informasi tentang frekuensi semua gelombang sinus yang diperlukan untuk menjumlahkan sinyal domain waktu.

Kata-kata kunci: amplitudo, sinyal arbitrer, Discrete Fourier Transform (DFT), frekuensi.

PENDAHULUAN

Pengolahan sinyal adalah suatu operasi matematik yang dilakukan terhadap suatu sinyal sehingga diperoleh suatu informasi yang dibutuhkan. Dalam hal ini terjadi suatu transformasi. (Elawati et al., 2020). Transformasi Fourier (Fourier Transform) merupakan salah satu metode populer yang sering digunakan untuk mengubah bentuk waktu bentuk frekuensi yang ditujukan untuk mempermudah analisis. Transformasi Fourier dibagi menjadi 2 bagian yaitu Fourier Kontinu dan Fourier Diskret. Untuk mengatasi berbagai permasalahan pada penggunaan Transformasi Fourier Kontinu maka mulai dikembangkan Discrete Fourier Transform (DFT) (Hidayaturrohmah et al., 2020). DFT bertujuan untuk membawa nilai sinyal kontinyu (analog) untuk dijadikan nilai diskrit agar dapat diproses atau dilakukan perhitungan secara digital menggunakan mikrokontroler atau komputer (Kusmiran, 2016).

Bentuk umum dari Discrete Fourier Transform dinyatakan dengan PERSAMAAN (1) dan (2) sebagai berikut (Ariyanto et al., 2018).

$$X(m) = \sum_{n=0}^{N-1} x(n)W_N^{mn} \quad (1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(m)W_N^{-mn} \quad (2)$$

Nilai $W_N = e^{-j(\frac{2\pi}{N})}$ dan $N = \text{Length}[x(n)]$

Keterangan:

$X(m)$ = Urutan ke (m) komponen output ($X(0), X(1), \dots, X(N-1)$)

m = Indeks output dalam domain frekuensi ($0, 1, \dots, N-1$)

$x(m)$ = Urutan ke (n) sampel input ($x(0), x(1), \dots, x(N-1)$)

n = Indeks sampel input dalam domain waktu ($0, 1, \dots, N-1$)

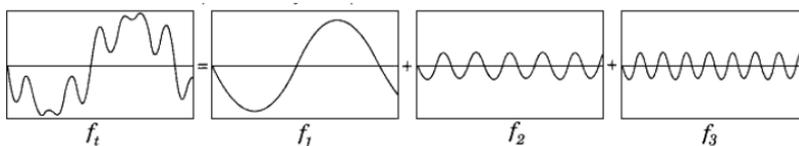
j = Bilangan imajiner ($\sqrt{-1}$)

π = Derajat (180°)

e = Bilangan logaritma natural (2,7182818)

1. Mengubah nilai sinyal diskrit pada domain waktu menjadi nilai diskrit dalam domain frekuensi

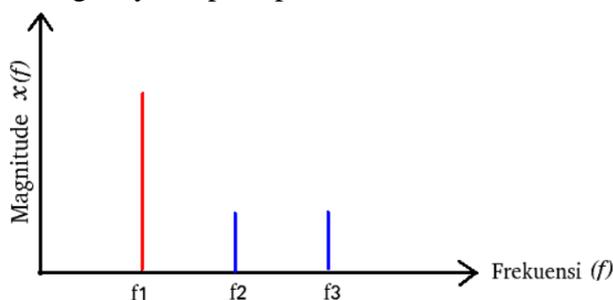
Sinyal pada domain waktu hanya dapat menunjukkan sebuah sinyal saja yang mungkin bentuk sinyal tersebut tampak tidak beraturan. Bentuk tidak beraturan tersebut (tidak murni sinusoidal) menandakan bahwa terdapat sinyal lain dengan frekuensi yang berbeda dan mungkin juga amplitudo atau magnitude yang berbeda mengganggu sebuah sinyal sinusoidal tersebut seperti ditunjukkan pada GAMBAR 1.



GAMBAR 1. Ilustrasi penjumlahan beberapa sinyal sinusoidal

2. DFT mampu mengetahui *magnitude* frekuensi-frekuensi pengganggu pada sebuah sinyal, atau mampu mengetahui *magnitude* banyak frekuensi pada suatu sinyal dimana hal ini sulit dilakukan pada domain waktu.

Hasil perhitungan DFT pada sinyal f_t GAMBAR 1 dapat membedakan atau memisahkan semua frekuensi yang berada pada sinyal f_t tersebut dengan menampilkan spektrum masing-masing sinyal seperti pada GAMBAR 2.



GAMBAR 2. Spektrum sinyal hasil perhitungan DFT

Dari penjelasan GAMBAR 1 dan 2 dapat disimpulkan bahwa DFT dapat mengubah variabel-variabel sinyal pada domain waktu menjadi variabel dalam domain frekuensi sehingga sebuah sinyal tidak beraturan pada domain waktu dapat diketahui semua frekuensi pembentuk sinyal tak beraturan tersebut. Variabel variabel sinyal pada domain frekuensi berupa magnitudo, sudut (fasa) dan frekuensi. (Adiwijaya et al., 2009).

METODE

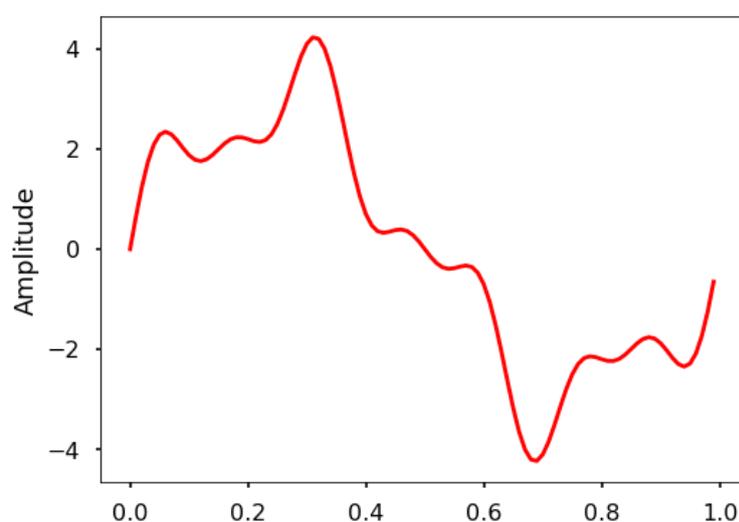
Program yang dikembangkan oleh peneliti yaitu, DFT untuk menganalisis sinyal arbitrer yang dibuat dalam dua program. Program pertama adalah fungsi dari DFT dan program kedua adalah program utama yang digunakan untuk memunculkan grafik DFT. Dalam project ini peneliti akan menganalisis sinyal arbitrer yang sudah diuraikan menjadi gelombang sinus. Setelah itu, untuk variasi peneliti akan menganalisis satu sinyal arbitrer lain yang sudah diuraikan juga menjadi gelombang sinus. Lalu, sebagai tambahan variasi, peneliti akan menggunakan `fft` dari `numpy` untuk perbandingan antara DFT dengan FFT lalu mencatat perubahan `time consumption` dari masing-masing metode untuk kedua sinyal yang akan dianalisis.

Program ini menggunakan dua library utama, yaitu `matplotlib` dan `numpy`. `numpy` adalah library python yang digunakan untuk bekerja dengan array dan juga memiliki fungsi yang bekerja dalam domain aljabar linier, transformasi fourier dan matriks (Shao et al., 2021). Lalu dari `numpy.fft` peneliti juga menggunakan library FFT agar dapat melakukan perhitungan FFT dari `numpy`. Kemudian, `matplotlib` adalah library yang bertanggung jawab untuk merencanakan data numerik. Library `Matplotlib` ini akan membantu dalam menampilkan hasil analisis berupa grafik berwarna dengan lebih rapi dan menarik (Ranjani et al., 2019).

HASIL DAN PEMBAHASAN

Berikut hasil output dari program penggunaan metode discrete fourier transform untuk menganalisis sinyal arbitrer

HASIL SINYAL ARBITRER DAN PLOT DFT



GAMBAR 3. Hasil output dari program penggunaan metode discrete fourier transform untuk menganalisis sinyal arbitrer

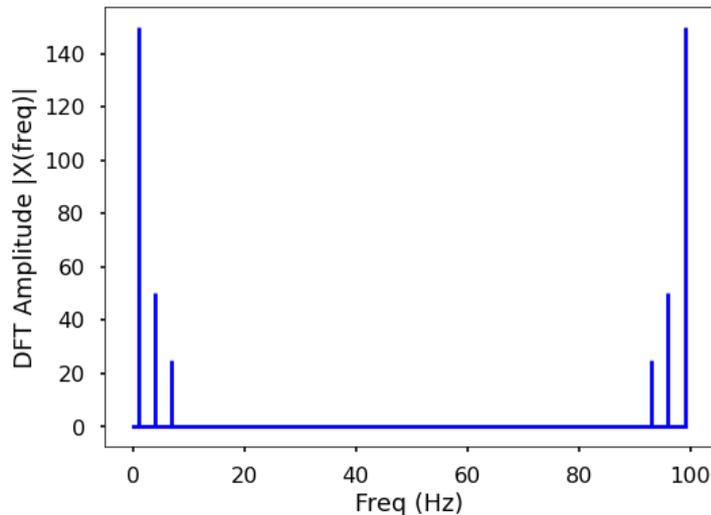
Sinyal tersebut sudah diuraikan menjadi sekumpulan gelombang sinus yang dapat diukur frekuensi, amplitude dan fasenya dengan mudah. Gelombang sinus tersebut merupakan kombinasi dari PERSAMAAN 3, 4, dan 5 gelombang sinus berikut:

$$x = 3 \sin 2\pi 1t \quad (3)$$

$$x = \sin 2\pi 4t \quad (4)$$

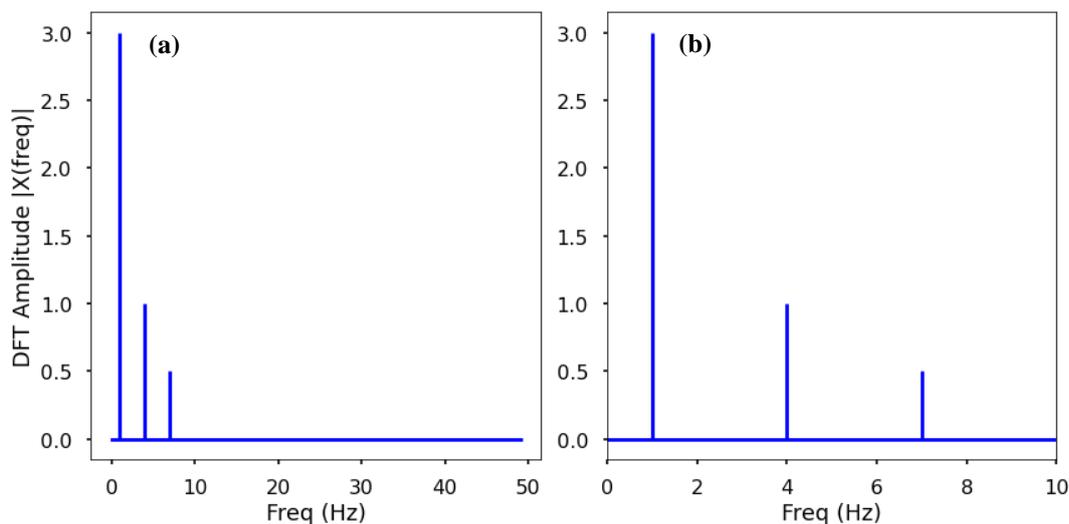
$$x = 0,5 \sin 2\pi 7t \quad (5)$$

Lalu, peneliti menghitung DFT dari sinyal tersebut. Berikut ini merupakan hasil plot yang didapat (GAMBAR 4).



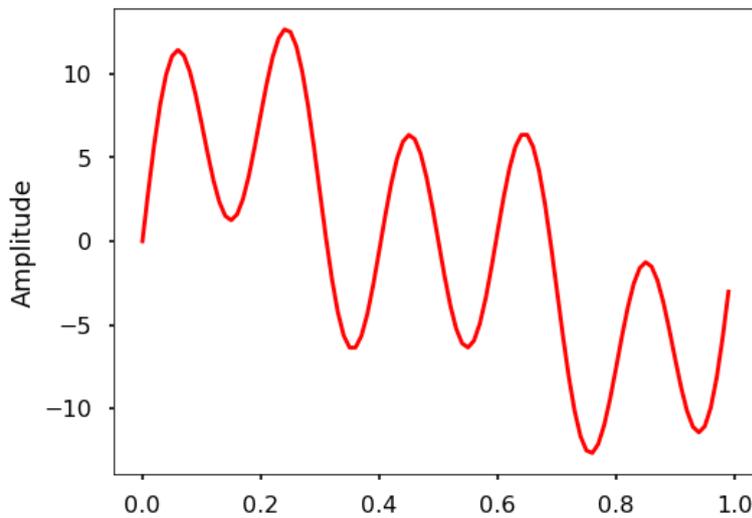
GAMBAR 4. Hasil plot DFT menggunakan sinyal arbitrer persamaan 3, 4, dan 5

Berdasarkan GAMBAR 4 karena hasil plot simetris pada setengah sampling rate, kesimpulan dapat dilihat dari salah satu sisi saja. GAMBAR 5 menampilkan hasil plot pada setengah sampling rate dan pada sampling rate 10 Hz. Hasil plot di GAMBAR 5 (a) merupakan hasil plot setengah sampling rate sedangkan hasil plot di GAMBAR 5 (b) merupakan hasil plot pada sampling rate 10 Hz.



GAMBAR 5. (a) Hasil plot DFT setengah sampling (b) Hasil plot DFT pada sampling rate 10 Hz

Adapun variasi dari sinyal arbitrer yang dianalisis ditampilkan pada GAMBAR 6:



GAMBAR 6. Variasi dari sinyal arbitrer yang digunakan pada plot DFT

Sinyal tersebut sudah diuraikan menjadi sekumpulan gelombang sinus yang dapat diukur frekuensi, amplitude dan fasenya dengan mudah. Gelombang sinus tersebut merupakan kombinasi dari tiga gelombang sinus pada PERSAMAAN 6, 7, dan 8.

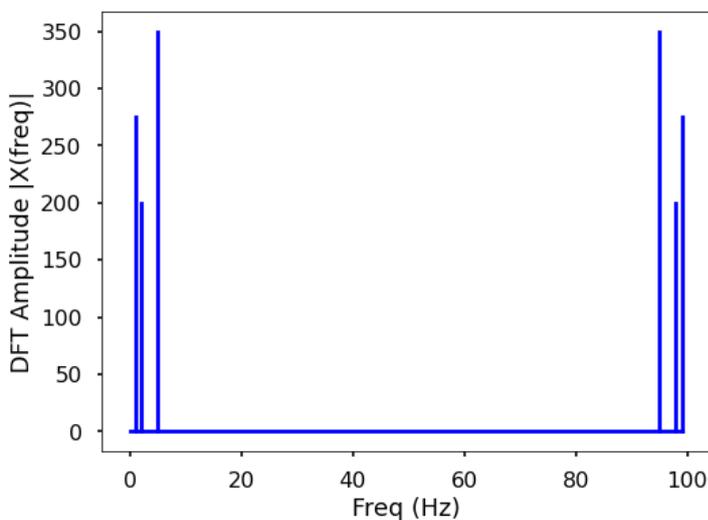
$$x = 7 \sin 2\pi 5t \quad (6)$$

$$x = 4 \sin 2\pi 2t \quad (7)$$

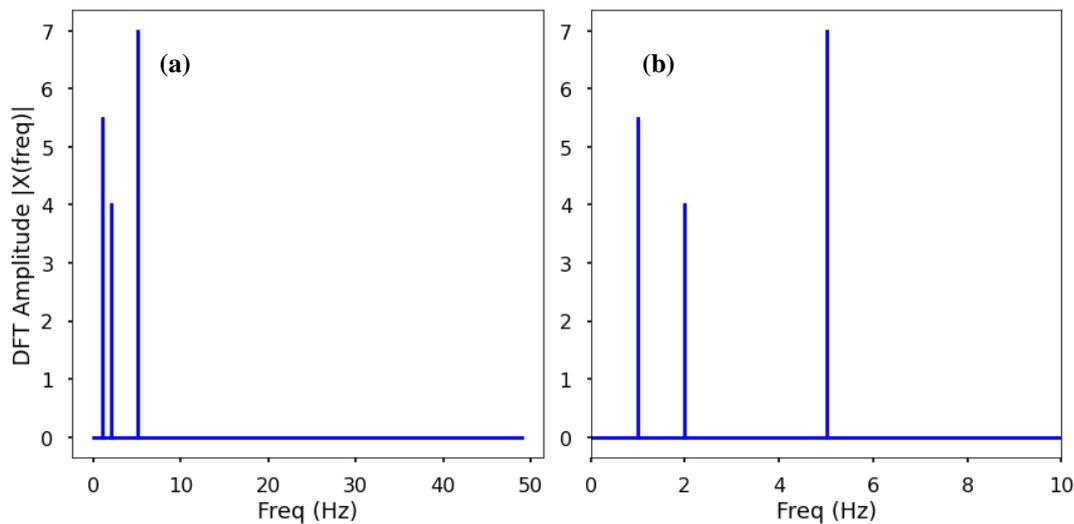
$$x = 5,5 \sin 2\pi 1t \quad (8)$$

Lalu, peneliti akan menghitung DFT dari sinyal tersebut. Berikut pada GAMBAR 7 merupakan hasil plot yang peneliti dapat.

Berdasarkan GAMBAR 7 karena hasil plot simetris pada setengah sampling rate, maka dapat ditarik kesimpulan dengan melihat di salah satu sisi saja. Berikut hasil plot pada setengah sampling rate GAMBAR 8 (a) dan pada sampling rate 10 Hz GAMBAR 8 (b).



GAMBAR 7. Hasil plot DFT menggunakan sinyal arbitrer persamaan 6, 7, dan 8



GAMBAR 8. (a) Hasil plot DFT setengah sampling (b) Hasil plot DFT pada sampling rate 10 Hz

PEMBAHASAN

Pada case kali ini, telah menggunakan metode Discrete Fourier Transform dan membuat hasil plot dari perhitungan yang dilakukan. Sekarang peneliti akan menganalisis proses yang peneliti lakukan selama menggunakan metode DFT dan membuat hasil plot.

Sebelum membahas hasil perhitungan, mari kembali membahas metode DFT itu sendiri. Dengan menggunakan DFT, dapat disusun sinyal arbitrer menjadi serangkaian sinusoid dan masing-masing akan memiliki frekuensi yang berbeda. Ide dibalik DFT itu sendiri bahwa sinyal arbitrer sebenarnya adalah hasil penjumlahan dari sejumlah gelombang sinus yang berbeda. Sinyal domain waktu dapat diubah menjadi gambar dalam domain frekuensi yang disebut spektrum amplitudo DFT, di mana frekuensi sinyal ditampilkan sebagai batang vertikal. Ketinggian bar setelah normalisasi adalah amplitudo sinyal dalam domain waktu. Peneliti memplot pada case kali ini adalah spektrum amplitudo DFT dari gelombang arbitrer yang sudah dilihat pada bagian sebelumnya.

Pertama, akan dibahas sinyal arbitrer yang berbentuk seperti GAMBAR 3. Sinyal arbitrer ini jika diuraikan menjadi gelombang sinus merupakan perpaduan dari gelombang sinus PERSAMAAN 3, 4, dan 5. Dapat dilihat pada persamaan di atas, PERSAMAAN (3) memiliki amplitudo = 3 dan frekuensi = 1, PERSAMAAN (4) memiliki amplitudo = 1 dan frekuensi = 4, serta PERSAMAAN (5) memiliki amplitudo = 0,5 dan frekuensi = 7. Dengan amplitudo dan frekuensi dari ketiga persamaan tersebut maka dapat diplot spektrum amplitudo dalam domain waktu dengan menghitung menggunakan metode DFT. Hasil plot yang didapat seperti GAMBAR 4.

Perlu diketahui bahwa sampling rate frekuensi yang diambil sampai dengan 100 Hz. Maka dari itu dapat dilihat pada grafik frekuensinya mencapai 100 Hz. Peneliti dapat melihat bahwa output DFT simetris pada setengah dari sampling rate. Setengah dari sampling rate ini disebut frekuensi Nyquist atau frekuensi lipat. Frekuensi Nyquist menyatakan bahwa sinyal sampel pada suatu laju dapat sepenuhnya direkonstruksi jika hanya berisi komponen frekuensi di bawah setengah sampling rate, sehingga keluaran frekuensi tertinggi dari DFT adalah setengah sampling rate.

Lalu diplot bagian setengah pertama hasil DFT dan peneliti dekatkan sampai sampling rate nya 10 Hz seperti pada GAMBAR 5. Dapat terlihat 3 puncak yang jelas pada frekuensi 1 Hz, 4 Hz, dan 7 Hz, dengan amplitude 3, 1, dan 0,5, seperti yang diharapkan. Setelah itu dapat terlihat variasi dari sinyal arbitrer yang akan dianalisis pada GAMBAR 6. Sinyal arbitrer ini jika diuraikan menjadi gelombang sinus merupakan perpaduan dari gelombang sinus PERSAMAAN 6, 7, dan 8. Dapat dilihat pada persamaan di atas, PERSAMAAN (6) memiliki amplitude = 7 dan frekuensi = 5, PERSAMAAN (7) memiliki amplitude = 4 dan frekuensi = 2, serta PERSAMAAN (8) memiliki amplitude = 5,5 dan frekuensi = 1. Dengan amplitude dan frekuensi dari ketiga persamaan tersebut dapat dibuat plot spektrum amplitude dalam domain waktu dengan menghitung menggunakan metode DFT. Hasil plot yang didapat disajikan pada GAMBAR 7. Karena output DFT simetris pada setengah dari sampling rate, peneliti bisa plot bagian setengah pertama hasil DFT dan peneliti dekatkan sampai sampling rate nya 10 Hz. Seperti yang diharapkan, output yang dihasilkan dapat terlihat 3 puncak yang jelas pada frekuensi 1 Hz, 2 Hz, dan 5 Hz, dengan amplitude 5,5, 4, dan 7.

Dari sisi algoritma, tidak ada yang berbeda jauh dari sinyal pertama dan sinyal variasi hanya berbeda di persamaan sinus, frekuensi, dan amplitudonya saja. Tetapi, terdapat perbedaan untuk metode DFT dan FFT di mana metode DFT menggunakan fungsi DFT yang peneliti inisiasikan sedangkan FFT menggunakan command FFT dari numpy. Dari sisi time consumption, waktu yang dibutuhkan untuk program berjalan tidak berbeda jauh. Untuk memplot grafik gelombang sinus sinyal pertama dan sinyal variasi dibutuhkan waktu 0,7s dan 0,4s. Untuk memplot hasil perhitungan DFT sinyal pertama dan sinyal variasi dengan sampling rate 100 Hz dibutuhkan waktu 0,9s dan 0,8s. Waktu peneliti dapat dari keterangan waktu running yang merupakan fitur dari Visual Studio Code.

Untuk mengukur time consumption dari metode DFT dan FFT peneliti menggunakan command python %timeit yang memberikan waktu eksekusi per loop. Dan untuk mengukur time consumption ini peneliti mengganti sample ratenya menjadi 2000 dan hanya akan memplot hasil plot pada sampling rate 50 Hz (sebelumnya setengah sampling rate dari 100 Hz) dan pada sampling rate 10 Hz. Peneliti melakukan 5 kali percobaan running dan mencatat rata-rata masing-masing time consumptionnya.

Hasil yang peneliti dapat untuk memplot metode DFT sinyal pertama yaitu $785,2 \text{ ms} \pm 105 \text{ ms}$ dalam 1 kali loop dan sinyal kedua yaitu $825,4 \text{ ms} \pm 128,5 \text{ ms}$ dalam 1 kali loop. Untuk memplot metode FFT sinyal pertama yaitu $114,1 \mu\text{s} \pm 23,7 \mu\text{s}$ dalam 10.000 kali loop dan sinyal kedua yaitu $93,3 \mu\text{s} \pm 20,2 \mu\text{s}$ dalam 10.000 kali loop.

Dapat dilihat dari hasil yang peneliti dapat, metode DFT dan FFT menghasilkan perbedaan time consumption yang sangat berbeda jauh. Ini membuktikan kekurangan metode DFT yang kurang efisien untuk melakukan perhitungan untuk sampling rate yang tinggi. Tentu saja waktu running ini dipengaruhi oleh banyak faktor salah satunya yaitu: spesifikasi computer dan banyaknya loop yang dilakukan (Lin et al., 2021).

Dari sisi cost-computation, kedua sinyal tidak memiliki perbedaan waktu yang signifikan kecuali antara metode DFT dan FFT dan outputnya menghasilkan ketelitian yang akurat dan sesuai dengan yang diharapkan. Karena peneliti melakukan running untuk sampling rate 2000, ini membutuhkan kekuatan komputasi yang cukup besar. Maka dari itu, spesifikasi computer juga berpengaruh dalam sisi cost-computation ini.

REFERENSI

- Adiwijaya, A., Suryandari, D. R., & Yulianto, F. A. (2009). Dekomposisi Nilai Singular dan Discrete Fourier Transform untuk Noise Filtering pada Citra Digital. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Ariyanto, Yuri., Ardiansyah, R., & Paris, B. 2018. Steganografi Menggunakan Metode Discrete Fourier Transform (DFT). *Jurnal Informatika Polinema*, 4(2), 87-92.
- Elawati, E., Hayati, R., & Hanafi, H. (2020). Analisis Perbandingan Metode Discrete Fourier Transform Dan Metode Discrete Cosine Transform Pada Teknik Menyembunyikan Sinyal Suara. *Jurnal TEKTR0*, 4(2), 82-89.
- Hidayaturrohman, U., Erfiani, E., & Afendi, F. M. (2020). Implementasi Transformasi Fourier Untuk Transformasi Domain Waktu Ke Domain Frekuensi Pada Luaran Purwarupa Alat Pendeteksian Gula Darah Secara Non-Invasif. *Indonesian Journal of Statistics and Its Applications*, 4(2), 234-244.
- Kusmiran, A. (2016). Implementasi Algoritma Discrete Furier Transform untuk Karakterisasi Nada dari Huruf Vokal. *Jurnal Tambora*, 1(2), 39-42.
- Lin, L., Xu, Z., Huan, H., Jian, Z., & Li-Xin, L. (2021). An Empirical Comparison of Implementation Efficiency of Iterative and Recursive Algorithms of Fast Fourier Transform. In *International Conference on Machine Learning and Intelligent Communications* (pp. 73-81). Springer, Cham.
- Ranjani, J., Sheela, A., & Meena, K. P. (2019, April). Combination of NumPy, SciPy and Matplotlib/Pylab-a good alternative methodology to MATLAB-A Comparative analysis. In *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)* (pp. 1-5). IEEE.
- Shao, X., Jiang, K., Mi, W., Genova, A., & Pavanello, M. (2021). DFTpy: An efficient and object-oriented platform for orbital-free DFT simulations. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 11(1), e1482.

LAMPIRAN 1. Algoritma program penggunaan metode discrete fourier transform untuk menganalisis sinyal arbitrer

Membuat fungsi DFT

1. Mendefinisikan DFT(x)
 - 1.1. Menginisiasi N, n, k, e, dan X
 - 1.2. Mengembalikan nilai X

Program Utama

1. Memulai program
2. Mengimport library matplotlib, numpy, dan fft dari numpy.fft
3. Menginisiasi style plot seaborn-poster
4. Menginisiasi sr, ts, dan t,
5. Menginisiasi $\text{freq} = 1$ dan $x = 3 * \text{np.sin}(2 * \text{np.pi} * \text{freq} * t)$
6. Menginisiasi $\text{freq} = 4$ dan $x += \text{np.sin}(2 * \text{np.pi} * \text{freq} * t)$
7. Menginisiasi $\text{freq} = 7$ dan $x += 0.5 * \text{np.sin}(2 * \text{np.pi} * \text{freq} * t)$
8. Menampilkan plot
9. Menginisiasi X, N, n, T, dan freq.
10. Menampilkan plot dengan freq sebagai sumbu x dan nilai absolut X sebagai sumbu y.
11. Menginisiasi $n_oneside = N // 2$
12. Menginisiasi $f_oneside = \text{freq}[n_oneside]$
13. Menginisiasi $X_oneside = X[n_oneside] / n_oneside$
14. Menampilkan plot untuk sampling rate 50 Hz dan 10 Hz
15. Mengukur waktu untuk mengeksekusi DFT(x) dan fft(x)