Perancangan Program *Python* Untuk Pengurangan *Noise* Pada Pengolahan Data Geomagnetik Menggunakan Metode *Moving Average* Dan FFT (*Fast Fourier Transform*)

Python Program Design for Noise Reduction in Geomagnetic Data Processing Using the Moving Average and FFT (Fast Fourier Transform) Methods

Yunita Sakira, Teti Zubaidah, Cipta Ramadhani
Jurusan Teknik Elektro, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram 83125, Lombok – Indonesia
E-mail: yunitasakira02@gmail.com

ABSTRAK

Medan magnet bumi selalu mengalami perubahan disebabkan oleh berbagai aktivitas di bumi atau luar bumi. Hasil pengukuran dari observatorium seringkali mengalami kerusakan data yang dipengaruhi oleh *noise* sehingga perlu diminimalisir. Data yang digunakan adalah data deret waktu variasi harian medan magnet total (F) dari Stasiun Nurul Bayan (NRB) pada bulan September 2018 yang terekam selama 24 jam dalam satuan detik. Penelitian ini menggunakan metode FFT untuk mengubah data dari domain waktu ke domain frekuensi dan metode *moving average* yang digunakan dalam merancang program *Python*. Berdasarkan hasil pengujian sistem menunjukkan, bahwa rancangan program *Python* dapat membersihkan *noise* yang dibuktikan dengan peningkatan nilai autokorelasi dari *rawdata* 0.054896 menjadi 0.999882 serta nilai SNR = 67.54 (dB) dan nilai MSE 0.0016 (dB).

Kata Kunci : Geomagnetik, *Noise*, FFT, *Moving Average, Program Python.*

ABSTRACT

The earth's magnetic field is always changing due to various activities on Earth or outside the Earth. Measurement results from observatories often experience data corruption, which is affected by noise and needs to be minimized. The data used is time series data of the daily variation of the total magnetic field (F) at Nurul Bayan Station (NRB) in September 2018, which was recorded for 24 hours in seconds. This study used the FFT method to convert data from the time domain to the frequency domain and the moving average method used in designing Python programs. Based on the results of system testing, it shows that the Python program design can clean noise, as evidenced by the increase in the autocorrelation value from 0.054896 to 0.999882 for raw data and the SNR of 67.54 dB and the MSE value of 0.0016 dB.
Keywords: geomagnetic, noise, FFT, moving average, Python program.

1. Pendahuluan

Medan magnet bumi yang bersumber dari dalam inti bumi dan kerak bumi relatif tetap sedangkan yang dari luar bumi yang selalu mengalami perubahan atau variasi setiap saat, baik variasi reguler maupun variasi tidak regular (Kamide dan Berekke, 1975). Metode yang dapat digunakan untuk menyelidiki kondisi bumi dengan memanfaatkan sifat kemagnetan bumi adalah metode geomagnetik. Metode geomagnetik berlandaskan pada pengukuran variasi intensitas magnetik di permukaan bumi diakibatkan adanya variasi distribusi benda termagnetisasi. Intensitas medan magnet bumi yang terekam pada magnetometer merupakan superposisi dari

berbagai sumber variasi, termasuk *noise* akibat interferensi variasi dari sumber lain baik eksternal maupun internal. *Noise* itu sendiri merupakan anomali yang muncul pada data dari suatu pengamatan dimana anomali tersebut bukan karakteristik dari pengamatan itu sendiri (Pranoto, 2016).

Sementara data geomagnetik yang terukur dengan magnetometer merupakan data dalam representasi domain waktu sehingga untuk dapat mengetahui keberadaan *noise* yang terukur tersebut, terlebih dahulu data *time series* ditransformasikan ke bentuk domain frekuensi dengan menggunakan metode FTT (*fast fourier transform*). *Fourier Transform* merupakan suatu metode yang pedigunakan untuk mengubah sinyal suatu gelombang dalam domain waktu menjadi domain frekuensi. Sedangkan untuk melakukan pengurangan *noise* yang mempengaruhi kualitas data, salah satu pendekatannya dengan menggunakan metode *moving average*. *Moving Average* adalah indikator yang menghitung harga rata-rata suatu data dalam periode waktu tertentu, kemudian menghubungkannya dalam bentuk garis. *Moving average* sering digunakan dalam deret waktu untuk menghaluskan variasi stokastik jangka pendek dan menyoroti komponen lain (tren, musim, atau siklus) dalam data (Moreno,2020).

Berdasarkan uraian permasalahan di atas, maka penulis tertarik untuk merancang suatu program berbasis *Python* sebagai media pengolahan data geomagnetik yang diharapkan mampu untuk mengetahui *noise* yang terkandung pada data geomagnetik menggunakan metode *moving average* dan metode FFT (*Fast Fourier Transform*).

2. Pengumpulan Data

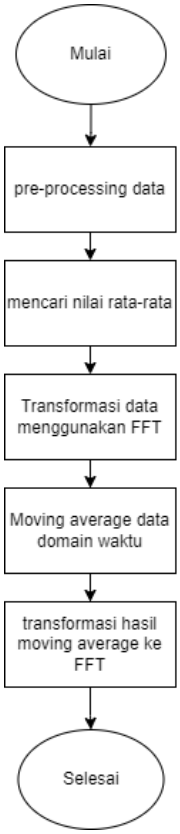
Alat dan bahan yang digunakan dalam penelitian ini terdiri dari data *time series*, perangkat keras dan perangkat lunak serta perangkat pendukung lainnya. Untuk data yang akan dianalisis adalahh data geomagnetik variasi harian medan magnet total (F) dari Stasiun Nurul Bayan (NRB) Lombok Utara milik Universitaas Mataram serta data yang telah terintegrasi dalam jaringan INTERMAGNET IAGA yaitu Observatorium Kakadu (KDU) Australia dan Observatorium Guam (GUA) United States of Amerika.

Data geomagnetik yang digunakan merupakan data yang terekam per detik selama 24 jam pada bulan September 2018. Data tersebut telah direpresentasikan dalam bentuk tabel yang disimpan dengan format *file* txt.

3. Metode penelitian

Rancangan program akan dibuat dalam bentuk program *python* dengan menggunakan *software* VScode. Untuk memulai tahap perancangan program *Python*, langkah awal yang harus dilakukan adalah menjalankan aplikasi VScode dan mengimport library yang dibutuhkan. Di dalam tahap pengolahan data mencakup data *pre-processing*, mencari nilai rata-rata, implesenrasi algoritma *fast fourier transform* dan *moving average*.

Cara kerja secara menyeluruh program diilustrasikan oleh diagram alir pada gambar 3.2 berikut ini.



Gambar 3.1 Diagram alir cara kerja sistem

1. Data Pre-processing

Sebelum data diproses pada program maka terlebih dahulu memasukkan atau ekspor file ke dalam program Python kemudian melakukan data pre-procesing yang mencakup format data, kelengkapan variabel data dan tipe data untuk melakukan pengolahan menggunakan program.

2. Mencari Nilai Rata-Rata

Pada tahap ini yaitu mencari nilia rata-rata, dimana tujuannya adalah agar data time series terorganisir atau dapat dibedakan antara data maksimum dan minimum. Berikut ini ditunjukkan perhitungan untuk mencari niali rata-rata.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} xi \qquad (3\text{-}1)$$

Keterangan:

$\bar{x}$ = rata-rata hitung

xi = nilai sample ke-i

n = jumlah sample

3. Fast Fourier Transform

Pada tahap ini, data domain waktu akan ditransformasi ke bentuk domain frekuensi menggunkan algoritma Fast Fourier Transform untuk mengetahui adanya noise yang ditunjukkan dengan plot data domain frekuensi. Dalam program Python sudah tersedia pustaka untuk melakukan proses transformasi Fourier tersebut, pada penelitian ini penulis menggunakan pustaka from scipy.fftpack import fft. Berikut ini merupakan fungsi fft yang akan digunakan pada script program nantinya.

```
yf = np.fft.fft()
```

Radix yang digunakan dalam program ditentukan berdasarkan prinsip FFT yaitu dengan memotong input (domain waktu) DFT menjadi beberapa bagian (2,4,8, dan seterusnya).

Bentuk umum dari Transformasi Fourier Diskrit dinyatakan dengan persamaan berikut.

$$X(m) = \sum_{n=0}^{N-1} x(n) W_N^{mn} \qquad (3\text{-}2)$$

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(m) W_N^{-mn} \qquad (3\text{-}3)$$

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

Keterangan:

$X(m)$: Urutan ke (m) komponen output ($X(0)$,$X(1)$, . . , $X(N-1)$)

m : Indeks output dalam domain Frekuensi (0, 1, . . , $N-1$)

$x(n)$: Urutan ke (n) sampel input ($x(0)$, $x(1)$, . . , $x(N-1)$)

n : Indeks sampel input dalam domain waktu (0, 1, … , $N-1$)

j : Bilangan Imajiner ($\sqrt{-1}$)

π : Derajat (1800)

e : Basis logaritma natural (2,7182818)

4. *Moving Average*

Pada tahap sebelumnya, setelah diketahui adanya *noise* pada data yang ditunjukkan dalam domain frekensi, maka langkah selanjutnya adalah menghilangkan *noise*. Pada tahap ini, *noise* tersebut akan dihilangkan menggunakan metode *moving average* pada data domain waktu. Setelah *noise* dihilangkan maka didapatkan data bersih atau bebas dari *noise*. Kemudian data domain waktu tersebut ditransformasi lagi ke bentuk domain frekunsi untuk dibandingkan dengan data yang mengandung *noise* dalam bentuk domain

frekunsi yang sebelumnya telah ditansformasi.

$$SMA = \frac{x_t + x_{t-1} + x_{t-2} + \cdots + x_{M-(t-1)}}{M} \quad (3\text{-}4)$$

Cara termudah untuk menghitung *simple Moving average* sederhana adalah dengan menggunakan metode pandas.Series.rolling.

1. *Signal to Noise Ratio* (SNR)

SNR (*Signal to Noise Ratio*) ialah perbandingan antara daya sinyal yang dinginkan dengan daya sinyal yang tidak dinginkan (*noise*) pada suatu titik ukur. SNR menyatakan kualitas sinyal informasi yang diterima pada sistem transmisi. SNR juga merupakan batas ambang sinyal *analog* yang masih dapat diterima. Semakin besar nilai SNR maka kualitas sinyal semakin bagus. (Pratama, dkk. 2020).

$$SNR = 10\ log_{10}\left[\frac{\frac{1}{n}\sum_{i=1}^{n} f^2(i)}{\frac{1}{n}\sum_{i=1}^{n}[f(i)-f(t)]^2}\right] \quad (3\text{-}5)$$

Daya sinyal yaitu *f(i)* dan sinyak reduksi atau noise yaitu *f(t)* dan panjang sinyal adalah n.

2. *Mean Squared Error* (MSE)

MSE adalah metode untuk mengukur tingkat keakuratan suatu model peramalan. Nilai MSE dapat dianalogikan sebagai varian ditambah dengan kuadrat bias dari suatu model dan MSE sangat baik dalam memberikan gambaran terhadap seberapa konsisten model yang dibangun. (Pratama, dkk. 2020).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(fi - yi)^2 \quad (3\text{-}6)$$

Sinyal asli sebagai sinyal *f(i)* dan sinyal peramalan adalah *y(i)*, dan panjang sinyal adalah n.

3. *Autocorrelation*

Autocorrelation atau autokorelasi adalah representasi matematis dari tingkat kesamaan anatara deret waktu tertentu dan versi tertinggal dari dirinya sendiri selama interval waktu yang berurutan. Secara konseptual mirip dengan korelasi antara dua deret waktu yang berbeda, tetapi autokorelasi menggunakan deret waktu yang sama.

$$DW = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T}(e_t - e_{t-1})^2} \quad (3\text{-}4)$$

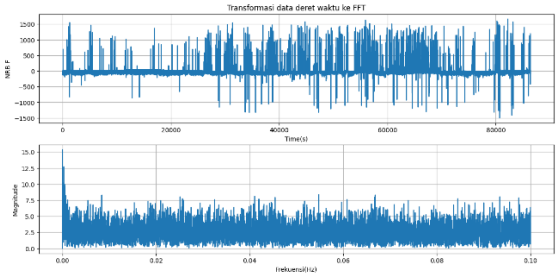
Di mana:

- et adalah angka residual
- T adalah jumlah pengamatan

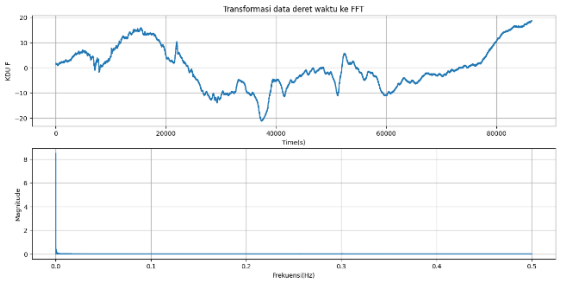
4. Hasil dan pembahasan

4.1 Fast Fourier Transform

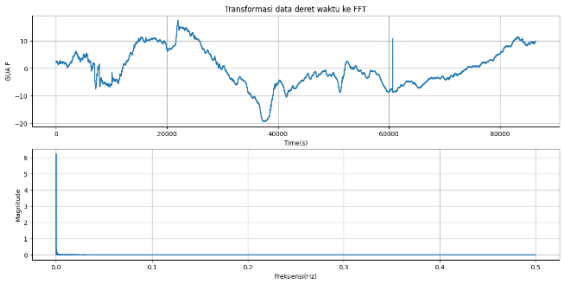
pada tahap ini data *time series* yang telah di input akan di ubah ke dalam domain frekuensi menggunakan algoritma FFT yang telah disediakan oleh pustaka *Python* yaitu *from scripy.fftpack import FFT.*



Gambar 4.1 Grafik transformasi data NRB dari domain waktu ke domain frekuensi



Gambar 4.2 Grafik transformasi data KDU dari domain waktu ke domain frekuensi



Gambar 4.3 Grafik transformasi data GUA dari domain waktu ke domain frekuensi

Pada Gambar di atas tersebut maka dapat dilihat perbandingan dari masing-masing observarotium yang menunjukkan bahwa gambar 4.1 terdapat perbedaan yang sangat signifikan terhadap data. Data yang telah terpengaruhi oleh gangguan atau

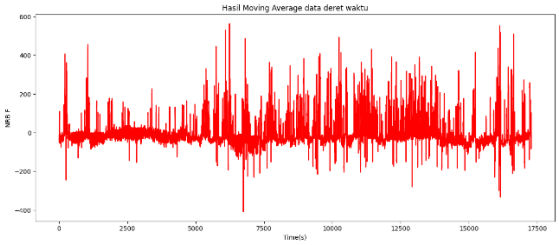
mengandung *noise* ditunjukkan dengan grafik variasi harian medan magnet total yang tidak beraturan (*scatter*). Rekaman data deret waktu sebanyak 17279 detik dengan variasi harian *raw data* yaitu 44831.21 nT dan selisih -97.99 nT. Kemudian ditransformasikan menggunakan FFT sehingga didapatkan nilai absolut 16.0 magnitudo dalam frekuensi 1 Hz. Berdasarkan pernyatan dari (Khomutov, dkk. 2017) data tersebut terpengaruhi oleh *noise* reguler dan *random noise* yang disebabkan karena dari masalah teknis dengan peralatan pengukuran atau interferensi dari perangkat lain yang beroperasi di sekitar magnetometer. Contoh indikatif dari *noise* biasa adalah interferensi pada data magnetik yang muncul saat *ionosonde* beroperasi.

Gambar 4.2 observatorium KDU menunjukkan bahwa variasi harian medan magnet total tidak mengandung *noise* sehingga menghasilkan grafik yang beraturan. Rekaman data deret waktu sebanyak 86400 detik dengan nilai variasi harian raw data yaitu 46184.99 nT dengan selisih 1.6 nT. Kemudian ditransformasikan menggunakan FFT sehingga didapatkan nilai absolut 9 magnitudo dalam frekuensi 0.5 Hz.

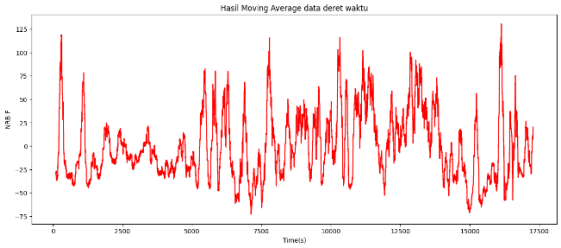
Gambar 4.3 observatorium GUA menunjukkan bahwa variasi harian medan magnet total tidak mengandung *noise* sehingga menghasilkan grafik yang beraturan. Rekaman data deret waktu sebanyak 86400 detik dengan nilai variasi harian dan raw data yaitu 36700.46 nT dengan selisih 2.5 nT. Kemudian ditransformasikan menggunakan FFT sehingga didapatkan nilai absolut 7 magnitudo dalam frekuensi 0.5 Hz.

4.2 Moving Average

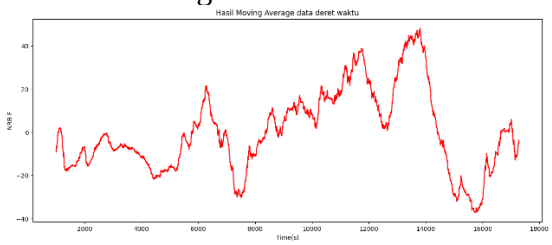
Pada tahap sebelumnya, setelah diketahui adanya kandungan data *noise* yang ditunjukkan dalam domain frekuensi pada gambar 4.13 maka langkah selanjutnya adalah menghilangkan *noise* dengan menerapkan metode moving average window 7, 100, 1000.



Gambar 4.4 *Moving average* data NRB dengan *window* 7



Gambar 4.5 *Moving average* data NRB dengan *window* 100



Gambar 4.6 *Moving average* data NRB dengan *window* 1000

Grafik 4.4 *raw data* deret waktu tersebut terlihat bentuk grafik yang masih tebal menandakan bahwa data tersebut masih mengandung *noise*. Nilai variasi harian dengan window 7 yaitu (-35.2873) nT pada (sumbu y), dimana data tersebut terekam selama 24 jam dalam satuan detik (sumbu x) dengan jumlah data sebanyak 17273 data.

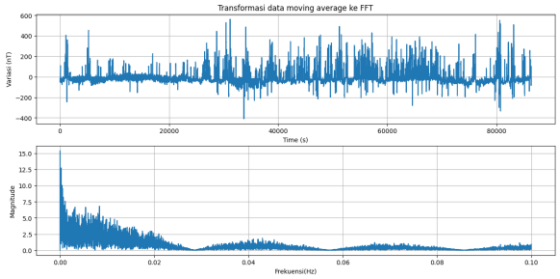
Grafik 4.5 *raw data* deret waktu tersebut terlihat lebih mulus dibandingkan dengan gambar 4.4 yang menandakan bahwa *noise* sudah berkurang. Nilai variasi harian dengan window 100 yaitu (-30.2322) nT pada (sumbu y),, dimana data tersebut terekam selama 24 jam dalam satuan detik (sumbu x) dengan jumlah data sebanyak 17180 data

Grafik 4.6 *raw data* deret waktu tersebut terlihat lebih mulus dibandingkan dengan gambar 4.5 yang menandakan bahwa *noise* sudah berkurang. Nilai variasi

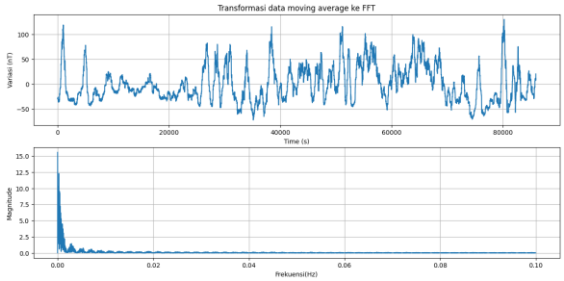
harian dengan window 1000 yaitu (-9.3485) nT pada (sumbu y), dimana data tersebut terekam selama 24 jam dalam satuan detik (sumbu x) dengan jumlah data sebanyak 16280 data. Dari perbandingan gambar di atas dapat dikatakan bahwa semakin besar *window* yang digunakan untuk melakukan *moving average* maka *noise* pun akan semakin berkurang hal ini ditandai dengan jumlah data *noise* yang telah hilang sehingga bentuk grafik semakin mulus dan nilai variasi yang semakin kecil yaitu pada gambar 4.2.6 menggunakan *window* 1000.

4.3 Analisa Perbandingan Grafik Domain Frekuensi Data NRB Hasil *Moving Average Window* 7, 100, dan 1000

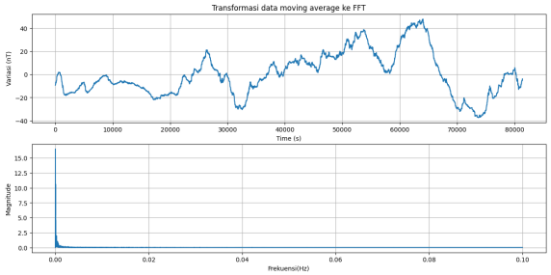
Berikut ini merupakan analisa perbandingan grafik domain frekuensi data NRB tanggal 5 September 2018 setelah diterapkan *moving average* dengan *window* 7, 100 dan 1000.



Gambar 4.7 Grafik domain frekuensi data NRB hasil *moving average window* 7



Gambar 4.8 Grafik domain frekuensi data NRB hasil *moving average window* 100



Gambar 4.9 Grafik domain frekuensi data NRB hasil *moving average window* 1000

Dari Gambar 4.7 dapat dilihat bahwa grafik *raw data* deret waktu tersebut terlihat bentuk grafik yang masih tebal menandakan bahwa data tersebut masih mengandung *noise*. Pada grafik domain waktu nilai selisih (-35.2873) nT pada (sumbu y). Rekaman data deret waktu sebanyak 17273 yang terekam selama 24 jam dalam satuan detik (sumbu x). Setelah itu data hasil *moving average* ditranformasikan menggunakan algoritma FFT ke dalam bentuk domain frekuensi sehingga dapat dilihat perbandingan data mengandung *noise* dan tidak mengandung *noise*. Pada gambar tersebut terdapat 4 komponen sinyal pada rentang frekuensi 1 Hz.

Gambar 4.8 grafik *raw data* deret waktu tersebut terlihat lebih mulus dibandingkan dengan Gambar 4.19 yang menandakan bahwa *noise* sudah berkurang. Pada grafik domain waktu nilai selisih (-30.2322) nT pada (sumbu y). Rekaman data deret waktu sebanyak 17180 yang terekam selama 24 jam dalam satuan detik (sumbu x). Kemudian data hasil *moving average* ditranformasikan menggunakan algoritma FFT ke dalam bentuk domain frekuensi sehingga dapat dilihat perbandingan data mengandung *noise* dan tidak mengandung *noise*. Pada gambar tersebut terdapat 4 komponen sinyal pada rentang frekuensi 1 Hz.

Gambar 4.9 grafik *raw data* deret waktu tersebut terlihat lebih mulus dibandingkan dengan gambar 4.20 yang menandakan bahwa *noise* sudah berkurang. Pada grafik domain waktu nilai selisih (-9.3485) nT pada (sumbu y). Rekaman data deret waktu sebanyak

17180 yang terekam selama 24 jam dalam satuan detik (sumbu x). Dari perbandingan gambar di atas dapat dikatakan bahwa semakin besar *window* yang digunakan untuk melakukan *moving average* maka *noise* pun akan semakin berkurang hal ini ditandai dengan jumlah data *noise* yang telah hilang sehingga bentuk grafik semakin mulus dan nilai variasi yang semakin kecil yaitu pada gambar 4.21 menggunakan *window* 1000.

4.4 Pengujian Sistem

Tujuan pengujian ini yaitu untuk menguji nilai SNR dan nilai MSE yang didapat setelah mereduksi *noise* menggunakan metode *moving average* dengan *window* yang berbeda. Pengujian ini adalah nilai rata-rata dari 17279 data NRB yang digunakan untuk pengujian.

Tabel 4.1 Hasil pengujian nilai SNR pada *raw data* NRB, KDU & GUA

Pengujian	Nilai SNR NRB	Nilai SNR KDU	Nilai SNR GUA
Raw data	47.28	74.83	74.51

Berdasarkan pada tabel di atas dapat diketahui nilai SNR pada *raw data* dari Stasiun Nurul Bayan yaitu 47.28 (dB) lebih kecil dibandingkan dengan nilai SNR observatorium pembanding yaitu observatorium KDU dengan nilai SNR 74.83 (dB) dan GUA dengan nilai SNR 74.51 (dB).

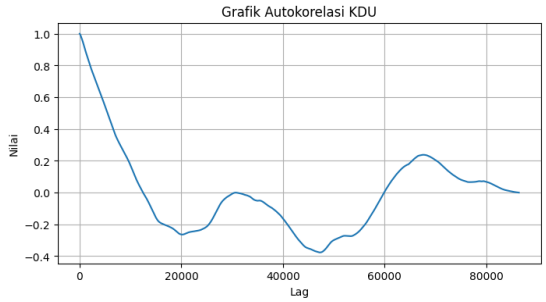
Tabel 4.2 Hasil pengujian *raw data* dengan data hasil *moving average*

Pengujian	Nilai SNR NRB (dB)	Nilai MSE (dB)
Raw data	47.18	-
Window 7	54.84	0.025
Window 100	61.94	0.019
Window 1000	67.54	0.0016

Berdasarkan pengujian pada tabel 4.2 dapat diketahui bahwa data NRB mengalami peningkatan nilai *Signal to Noise Ratio* SNR setelah dilakukan pengurangan *noise* menggunakan metode *moving average* dengan pemilihan jendela (*window*). Nilai SNR *window* 7 yaitu 54.84 (dB) dengan nilai MSE 0.025 (dB). Kemudian nilai SNR *window* 100 mengalami peningkatan 61.94 (dB) dan nilai MSE 0.019 (dB). Sedangkan peningkatan nilai SNR *window* 1000 mengalami peningkatan menjadi 67.54 (dB) dan nilai MSE 0.0016 (dB).

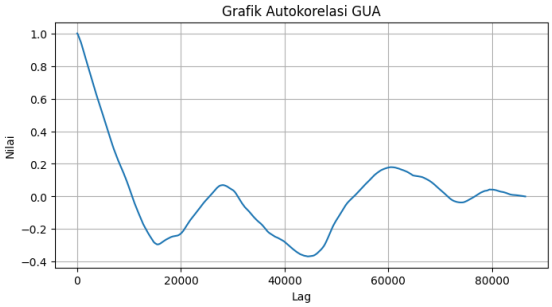
4.4.1 Pengujian Autokorelasi

Pengujian autokorelasi ini dilakukan untuk mengetahui seberapa erat hubungan antara nilai-nilai pada sebuah deret waktu. Dengan mengimplementasikan persamaan (3-4) maka didapatkan hasil sebagai berikut.



Gambar 4.10 grafik autokorelasi observatorium KDU

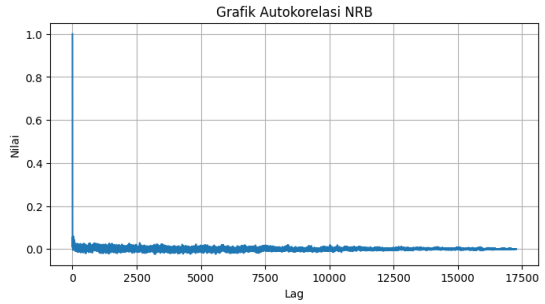
Berdasarkan grafik autokorelasi di atas dapat dilihat bahwa data observatorium yang diukur selama 24 jam dengan data sebanyak 86400, menunjukkan bahwa data mempunyai nilai autokorelasi sebanyak 45%.



Gambar 4.11 grafik autokorelasi observatorium GUA

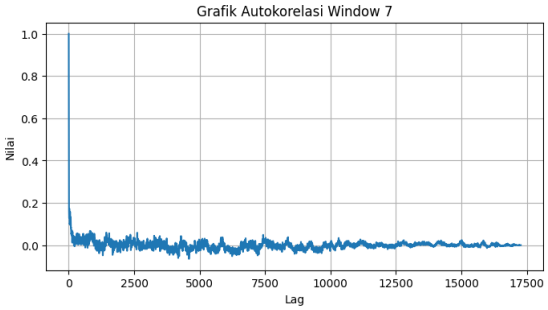
Berdasarkan grafik autokorelasi di atas dapat dilihat bahwa data observatorium yang diukur selama 24 jam

dengan data sebanyak 86400, menunjukkan bahwa mempunyai nilai autokorelasi sebanyak 50%.



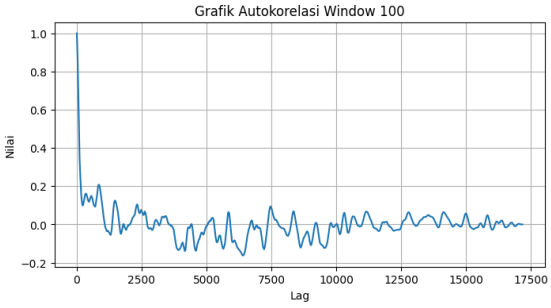
Gambar 4.12 grafik autokorelasi *raw data* Statiun Nurul Bayan

Berdasarkan grafik autokorelasi di atas dapat dilihat bahwa data observatorium yang diukur selama 24 jam dengan data sebanyak 17279 data, menunjukkan bahwa data mempunyai nilai autokorelasi sebanyak 44% namun dengan nilai autokorelasi yang sangat kecil hampir mendekati nilai 0.0.



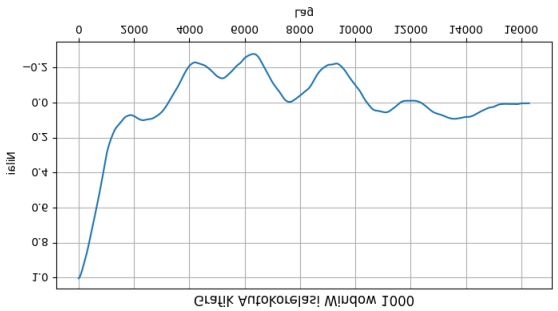
Gambar 4.13 grafik autokorelasi data *window* 7

Berdasarkan grafik autokorelasi di atas dapat dilihat bahwa data observatorium yang diukur selama 24 jam dengan data sebanyak 17273 data, menunjukkan bahwa data mempunyai nilai autokorelasi sebanyak 46%.



Gambar 4.14 grafik autokorelasi data *window* 100

Berdasarkan grafik autokorelasi di atas dapat dilihat bahwa data observatorium yang diukur selama 24 jam dengan data sebanyak 17180 data, menunjukkan bahwa mempunyai nilai autokorelasi sebanyak 47%.



Gambar 4.15 grafik autokorelasi data *window* 1000

Berdasarkan grafik autokorelasi di atas dapat dilihat bahwa data observatorium yang diukur selama 24 jam dengan data sebanyak 16280, menunjukkan bahwa data mempunyai nilai autokorelasi sebanyak 50%.

5. Kesimpulan dan Saran

5.1 kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil implementasi metode *Fast Fourier Transform* ke dalam program yang dirancang untuk mengubah data domain waktu ke dalam domain frekuensi maka dapat diketahui pada domain frekuensi nilai spektrum dan nilai variasi harian data NRB lebih besar dari observatorium KDU & GUA. Variasi harian raw data NRB yaitu 44831.21 nT dengan selisih -97.99 nT sedangkan raw data KDU yaitu 46184.99 nT dengan selisih 1.6 nT dan raw data GUA yaitu 36700.46 nT dengan selisih 2.5 nT. Berdasarkan klasifikasi *noise*, raw data NRB mengandung *noise* yang disebabkan oleh *spikes* dan masalah teknis dengan peralatan atau interferensi dari perangkat lain yang beroperasi di sekitar magnetometer.

2. Pengurangan *noise* dapat dilakukan dengan menerapkan metode *moving average* dengan pemilihan ukuran jendela (*window size*) berdasarkan refrensi klasifikasi *noise* yang terkandung. Berdasarkan hasil pengujian *Signal to Noise Ratio* (SNR) didapatkan hasil pada *window* 7 dengan nilai SNR 54.84 (dB) dan nilai MSE 0.025 (dB), dan *window* 100 dengan nilai SNR 61.94 (dB) dan nilai MSE 0.019 (dB), sedangkan pada *window* 1000 dengan nilai SNR 67.54 (dB) dan nilai MSE 0.0016 (dB), sehingga dapat dikatakan bahwa semakin meningkat nilai SNR semakin kecil daya *noise.*
3. Berdasarkan hasil uji autokorelasi *raw data* NRB dengan data tidak mengandung noise yaitu KDU dan GUA, rancangan program *python* ini dapat digunakan untuk membersihkan *noise* pada data NRB dibuktikan dengan peningkatan nilai autokorelasi dari *raw data* yang semulanya 0.054896 atau nilai autokorelasi kecil menjadi 0.999882 (window 1000) atau mendekati nilai autokorelasi observatorium pembanding yaitu 0.999993 (KDU) dan 0.999879 (GUA).

5.2 Saran

Berdasarkan hasil dari penelitian yang telah dilakukan, penulis mempertimbangkan potensi dan kekurangan yang dapat menjadi saran untuk penelitian selanjutnya, yaitu sebagai berikut:

1. Penambahan parameter untuk mengevaluasi jenis *noise* yang terkandung pada data sehingga dapat mengidentifikasi komponen sinyal lainnya.
2. Pengujian untuk mengolah data yang berkaitan dengan fenomena fisis sehingga dapat mengetahui komponen sinyal yang berkaitan dengan kejadian yang mempengaruhi aktivitas geomagnetik bumi seperti gempa bumi, interaksi angin surya dan bagai magnetik.

Daftar Pustaka

[1] Khomutov, S. Y., Mandrikova, O. V., Budilova, E. A., Arora, K., & Manjula, L. (2017). Noise in raw data from magnetic observatories. *Geoscientific Instrumentation Methods and Data Systems*.

[2] Manjula, L., Nelapatla, P.C. & Arora, K. (2022). Evaluating the Effect of Noise from Traffic on HYB Magnetic Observatory Data during COVID-19 Lockdown. *applied sciences*.

[3] Moreno, A. I. (2020). *Moving average with python*. Retrieved from Towards Data Science:https://towardsdatascience.com/moving-averages-in-python-16170e20f6c.

[4] Pranoto, S. C. (2016). Pemisahan Sinyal Noise Pada Pengolahan Data Medan Magnet bumi Menggunakan Transformasi Wavelet. In S. C. Pranoto, *PROSIDING SKF 2016* (p. 414). Jl. DR. Djundjunan no. 133 Bandung, Indonesia, 40173: Pusat Sains Antariksa.

[5] Pratama, Y. A., Novianty, A., & Prasasti, A. L., (2020). Noise Handling Pada Sinyal Seismik Menggunakan Fourier Transform. *e-Proceeding of Engineering : Vol.7, No.2 Agustus 2020 | Page 4699*, 4699.