

## LAPORAN PRAKTIKUM

# SAMPLING, FRAME BLOCKING DAN KONVOLUSI

Muhammad Fadhil Syahputra, Ismail Ragi Alfarugi .

Universitas Lambung Mangkurat

Email korespondensi : syahputraf112@gmail.com

## PENDAHULUAN

DDalam era digital saat ini, pengolahan sinyal menjadi aspek krusial dalam berbagai bidang teknologi, terutama yang berkaitan dengan komunikasi, audio, dan sistem kendali. Sinyal sebagai representasi fisik dari informasi dapat berupa sinyal analog maupun digital. Pengolahan sinyal analog dilakukan langsung terhadap bentuk gelombang kontinu menggunakan rangkaian analog seperti penguat atau filter. Namun, seiring perkembangan teknologi digital, pengolahan sinyal kini lebih banyak dilakukan secara digital karena keunggulannya dalam hal fleksibilitas, ketahanan terhadap gangguan, serta kemudahan dalam penyimpanan dan transmisi data. Agar sinyal analog dapat diproses secara digital, diperlukan proses digitalisasi yang terdiri dari tahapan sampling, kuantisasi, dan pengkodean menggunakan perangkat Analog-to-Digital Converter (ADC). Setelah diproses oleh prosesor digital, sinyal kemudian dapat dikembalikan ke bentuk analog melalui Digital-to-Analog Converter (DAC). Praktikum ini difokuskan pada pemahaman dasar mengenai proses sampling sebagai tahap awal pengolahan sinyal digital, serta pengenalan konsep frame blocking dan konvolusi yang menjadi landasan dalam pemrosesan lanjutan. Pemahaman lebih lanjut tentang kuantisasi dan pengkodean sangat disarankan untuk dikuasai secara mandiri, terutama bagi mahasiswa yang tertarik untuk mengeksplorasi aplikasi pengolahan sinyal digital pada bidang komunikasi atau kompresi sinyal

## TINJAUAN PUSTAKA

Sinyal merupakan arus data atau representasi fisik dari informasi yang bergerak dalam suatu sistem, umumnya melalui media transmisi seperti kabel, udara, atau serat optik. Bidang tertentu seperti sistem elektronik dan komunikasi menggunakan sinyal dalam

bentuk tegangan listrik, arus listrik, atau kuat medan elektromagnetik yang berubah-ubah sesuai dengan informasi yang dibawanya. Perubahan ini terjadi seiring waktu ( $t$ ), ruang, atau variabel-variabel lain sehingga sinyal seringkali didefinisikan sebagai besaran fisik yang memiliki nilai yang berubah terhadap suatu variabel bebas. Sinyal berfungsi sebagai media utama untuk mentransmisikan informasi dari satu titik ke titik lainnya, baik dalam bentuk suara, data, gambar, maupun video. Sinyal dalam bidang komunikasi sangat dibutuhkan dan vital karena menjadi dasar kerja berbagai perangkat seperti telepon seluler, radio, televisi, komputer, dan sistem navigasi. Tanpa sinyal yang stabil sistem komunikasi tidak mampu menjalankan fungsinya dengan baik (Khairunnisa, 2019).

Salah satu metode yang penggunaannya mendekomposisi suatu gelombang seismik menjadi beberapa gelombang harmonik sinusoidal (deret fourier) disebut dengan transformasi fourier. Transformasi Fourier merupakan salah satu metode matematika paling fundamental dalam analisis sinyal yang memungkinkan perubahan bentuk sinyal dari domain waktu ke domain frekuensi, sehingga komponen-komponen frekuensi dari sinyal dapat dianalisis secara terpisah. Manfaat transformasi fourier adalah untuk mengidentifikasi karakteristik sinyal seperti amplitudo, fase, dan spektrum frekuensi. Fast Fourier Transform (FFT) adalah contoh algoritma yang digunakan untuk menghitung transformasi Fourier diskrit (DFT) dengan mengeluarkan amplitudo getaran sebagai fungsi frekuensi untuk mengetahui penyebab terjadinya getaran (Suprpto & Yandra, 2021).

Pengolahan sinyal merupakan operasi matematik bagi suatu sinyal untuk memperoleh informasi dengan melibatkan terjadinya suatu transformasi. Spektrum sinyal

menggambarkan tingkat energi dalam sinyal sebagai fungsi frekuensi (Elawati et al., 2020). Spektrum frekuensi sinyal merupakan representasi suatu sinyal dalam domain frekuensi yang menggambarkan bagaimana energi atau informasi dalam sinyal tersebar terhadap frekuensi. Spektrum sinyal umumnya terdiri dari dua komponen utama

### METODE PENELITIAN

Metode percobaan ini menggunakan bahasa pemrograman Python untuk memvisualisasikan dan memahami konsep dasar pengolahan sinyal digital, yang mencakup proses sampling, frame blocking, dan konvolusi. Percobaan dimulai dengan membangkitkan sinyal analog buatan, seperti gelombang sinusoidal, menggunakan fungsi numpy untuk mensimulasikan sinyal kontinu. Proses sampling dilakukan dengan mengambil nilai-nilai sinyal pada interval waktu tertentu yang ditentukan oleh frekuensi sampling. Visualisasi sinyal sebelum dan sesudah sampling ditampilkan menggunakan pustaka matplotlib.

Setelah proses sampling, data hasil sampling akan diproses lebih lanjut dengan membaginya menjadi beberapa segmen menggunakan teknik frame blocking. Proses ini dilakukan dengan membagi sinyal diskrit

### HASIL DAN PEMBAHASAN

Tabel 1. Dasar Pembentukan FFT

No	Code	Gambar
1	<pre> T = 0.01  n = np.linspace(0, 200, 200) t=n*T x=3*np.sin(2*np.pi*t ) fig,      ax = plt.subplots(1,  2, figsize=(12, 6)) ax[0].plot(t,      x, color = next(colors)) ax[0].set_title('Sin e Wave Continuous') ax[0].set_xlabel('Ti me (s)') ax[0].set_ylabel('Am plitude') </pre>	

yaitu amplitudo dan fase. Kedua komponen ini masing-masing merepresentasikan besar kontribusi dan pergeseran sudut dari tiap-tiap frekuensi dalam menyusun sinyal tersebut. Spektrum frekuensi sinyal ini berguna dalam analisis lanjutan seperti filterisasi, kompresi, modulasi, hingga deteksi gangguan atau noise (Sopandi, 2006)

ke dalam frame berdurasi tetap dengan atau tanpa tumpang tindih (overlap), menggunakan perulangan atau fungsi khusus. Kemudian, setiap frame dapat diproses dengan operasi konvolusi, yaitu operasi linier yang sering digunakan dalam filtrasi sinyal. Proses konvolusi ini dilakukan dengan menggunakan fungsi numpy.convolve() terhadap sinyal dengan kernel (filter) tertentu, seperti low-pass filter atau high-pass filter sederhana.

Seluruh rangkaian proses, mulai dari pembuatan sinyal, sampling, frame blocking, hingga konvolusi, diimplementasikan dan divisualisasikan dalam Python. Dengan pendekatan ini, mahasiswa dapat memahami setiap tahapan pengolahan sinyal digital secara interaktif dan mendalam melalui simulasi dan analisis kode yang dijalankan secara langsung.

```

stemlines,
markerlines,
baselines =
ax[1].stem(t, x)
stemlines.set_color(
next(colors))
ax[1].set_title('Sin
e Wave Discrete')
ax[1].set_xlabel('Ti
me (s)')
ax[1].set_ylabel('Am
plitude')
ax[1].grid()
ax[0].grid()
plt.tight_layout()
plt.show()

```

```

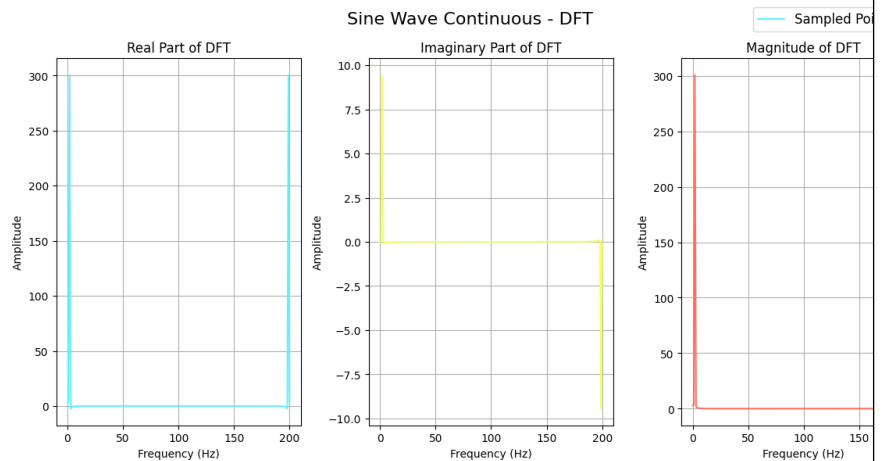
def dft(x: list) -> tuple:
    yreals = []
    yimags = []
    ymags = []
    N = len(x)
    for k in range(N):
        xn = 0
        for n in range(N):
            xn += x[n] * np.exp(-1j *
2 * np.pi * k * n / N)
        yreal = np.real(xn)
        yimag = np.imag(xn)
        ymag = np.sqrt(yreal**2 +
ymag**2)
        yreals.append(yreal)
        yimags.append(yimag)
        ymags.append(ymag)
    return np.array(yreals),
np.array(yimags),
np.array(ymags)

```

```

def plot_dft(x: list, subtitle: str)
-> None:
    sample = f'Sampled Points:
{len(x)}"
    fig, ax = plt.subplots(1, 3,
figsize=(12, 6))
    fig.suptitle(f'{subtitle} - DFT',
fontsize=16)
    yreals, yimags, ymags = dft(x)
    ax[0].plot(np.linspace(0,
len(yreals), len(yreals)), yreals,
color = next(colors))
    ax[0].set_title('Real Part of
DFT')
    ax[0].set_xlabel('Frequency
(Hz)')
    ax[0].set_ylabel('Amplitude')
    ax[1].plot(np.linspace(0,

```



	<pre> len(yimages), len(yimages)), yimages, color = next(colors)) ax[1].set_title('Imaginary Part of DFT') ax[1].set_xlabel('Frequency (Hz)') ax[1].set_ylabel('Amplitude') ax[2].plot(np.linspace(0, len(yimages), len(yimages)), ymags, color = next(colors)) ax[2].set_title('Magnitude of DFT') ax[2].set_xlabel('Frequency (Hz)') ax[2].set_ylabel('Amplitude') ax[2].grid() ax[0].grid() ax[1].grid() fig.legend([sample], loc='upper right', fontsize=12, frameon=True) plt.tight_layout() plt.show() </pre>	
3	<pre> T = 0.01  for N in range(200,1200, 200):     n = np.linspace(0, N, N)     t = n*T     x = 3*np.sin(2*np.pi*t)     plot_dft(x, 'Sine Wave Discrete')     x = 3*np.cos(2*np.pi*t)     plot_dft(x, 'Sine Wave Discrete - Cosine') </pre>	
4	<pre> for n in range(1, 16):     if n &lt;= 4:         x = np.append(x, 1)     else:         x = np.append(x, 0)  t= np.arange(1, len(x)+1, 1) fig, ax = plt.subplots(1, 2, figsize=(12, 6))  stemlines, markerlines, baselines = ax[0].stem(t, x) stemlines.set_color(next(colors) ) ax[0].set_title('Step Function Discrete') ax[0].set_xlabel('Time (s)') ax[0].set_ylabel('Amplitude') ax[0].grid()  s = np.fft.fft(x) spec = np.abs(s) </pre>	

```
f_x = np.arange(0, len(spec), 1)
ax[1].plot(f_x, spec, color =
next(colors))
ax[1].set_title('Step Function
Discrete')
ax[1].set_xlabel('Frequency
(Hz)')
ax[1].set_ylabel('Amplitude')
ax[1].grid()
plt.tight_layout()
plt.show()
```

## PEMBAHASAN

Percobaan ini berhasil memperlihatkan tahapan awal pengolahan sinyal digital, dimulai dari proses sampling hingga konvolusi. Melalui penggunaan Python dan pustaka seperti numpy dan matplotlib, sinyal analog dalam bentuk gelombang sinusoidal dapat direpresentasikan secara visual sebagai sinyal kontinu. Ketika dilakukan proses sampling dengan frekuensi tertentu, tampak bahwa sinyal kontinu berubah menjadi sinyal diskrit yang hanya terdiri dari titik-titik data pada waktu tertentu. Hal ini menunjukkan pentingnya pemilihan frekuensi sampling yang sesuai, di mana jika frekuensi sampling terlalu rendah (di bawah dua kali frekuensi sinyal), maka terjadi aliasing, yaitu penyimpangan bentuk sinyal yang disampling.

Tahap berikutnya, yaitu frame blocking, dilakukan untuk memecah sinyal diskrit ke dalam blok-blok kecil (frame) guna memudahkan proses pengolahan lanjutan. Dari percobaan, terlihat bahwa pemilihan ukuran frame dan langkah (stride atau hop size) mempengaruhi hasil pemrosesan, terutama dalam konteks efisiensi dan akurasi analisis data sinyal.

Pada proses konvolusi, hasil percobaan menunjukkan bahwa sinyal input mengalami transformasi tergantung pada kernel filter yang digunakan. Misalnya, jika digunakan kernel rata-rata (moving average filter), maka sinyal menjadi lebih halus karena komponen frekuensi tinggi tereduksi. Sebaliknya, penggunaan kernel perbedaan dapat menonjolkan perubahan tajam dalam sinyal, yang relevan untuk deteksi tepi atau perubahan mendadak.

Secara keseluruhan, percobaan ini memberikan pemahaman praktis dan visual terhadap konsep dasar pengolahan sinyal digital. Penggunaan Python memungkinkan eksplorasi fleksibel terhadap parameter-parameter penting seperti frekuensi sampling,

ukuran frame, dan bentuk kernel konvolusi, sehingga mahasiswa dapat memahami dampaknya terhadap karakteristik sinyal secara menyeluruh. Hal ini menjadi landasan penting bagi aplikasi lanjutan seperti pengenalan suara, sistem komunikasi digital, maupun kompresi data.

**DAFTAR PUSTAKA**

Elawati, E., Hayati, R., & Hanafi, H. (2020). Analisis Perbandingan Metode Descrete Fourier Transform dan Metode Descrete Cosine Transform pada Teknik Menyembunyikan Sinyal Suara. Jurnal TEKTR0, 4(2), 82-89.

Khairunnisa. (2019). Pengolahan Sinyal. Poliban Press, Yogyakarta.

Suprpto, E., & Yandra, F. E. (2021). Studi Analisis Spektrum Gelombang Petir dengan Menggunakan Fast Fourier Transform. Jurnal Manajemen Pendidikan dan Ilmu Sosial, 2(2), 889-897.

Sopandi, D. (2006). Instalasi dan Konfigurasi Jaringan Komputer. Informatika, Bandung.