2.1

```
import numpy as np
import matplotlib.pyplot as plt

# Amplitudo
A = 8
N = 4
n = np.arange(N)

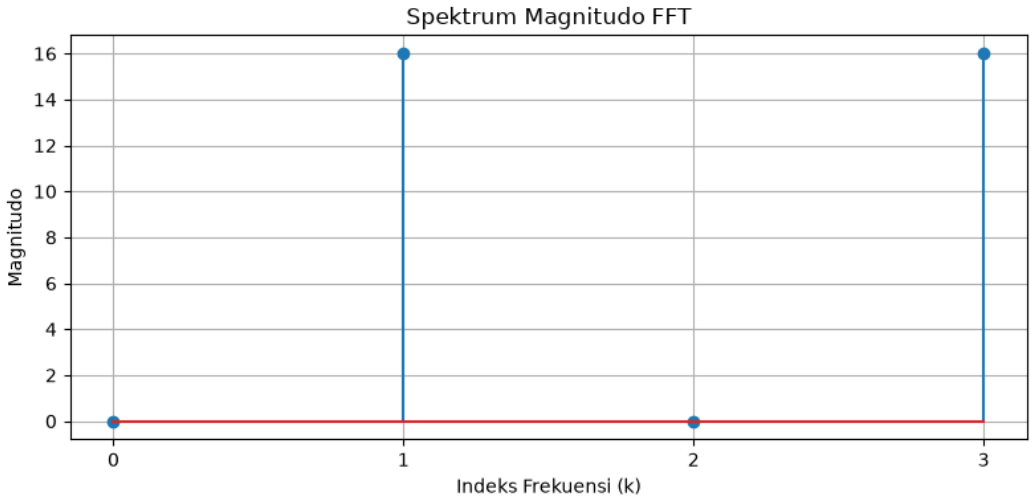
# Sinyal domain waktu
x = A * np.sin(2 * np.pi * (1/4) * n)

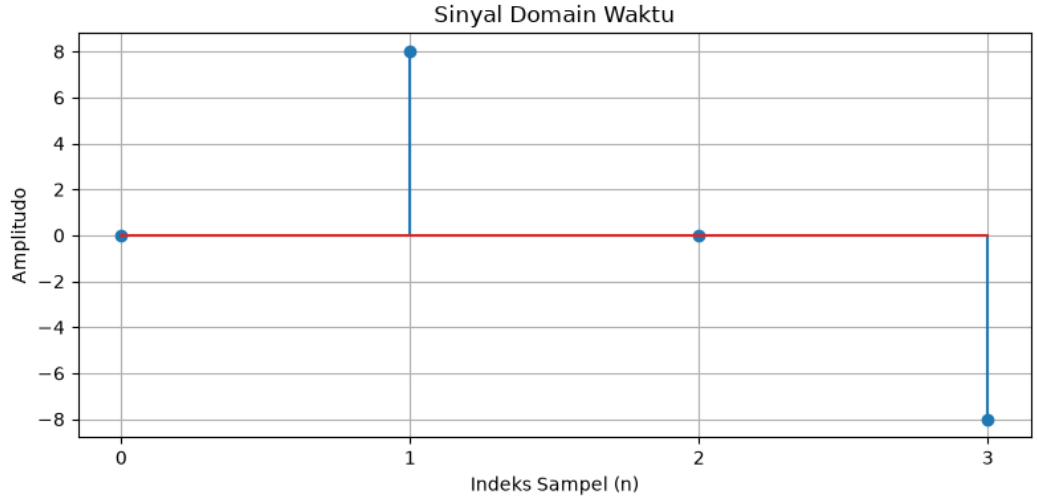
# Menghitung FFT
X = np.fft.fft(x)

print("Amplitudo:", A)
print("Sinyal domain waktu:", np.round(x, 4))

print("\nHasil FFT:")
for k in range(N):
    print(f"X[{k}] = {X[k].real:.4f} {X[k].imag:+.4f}j")

# Grafik sinyal domain waktu
plt.figure(figsize=(8, 4))
plt.stem(n, x)
plt.title("Sinyal Domain Waktu")
plt.xlabel("Indeks Sampel (n)")
plt.ylabel("Amplitudo")
plt.xticks(n)
plt.grid(True)
plt.tight_layout()
plt.show()
# Grafik spektrum magnitudo FFT
plt.figure(figsize=(8, 4))
plt.stem(np.arange(N), np.abs(X))
plt.title("Spektrum Magnitudo FFT")
plt.xlabel("Indeks Frekuensi (k)")
plt.ylabel("Magnitudo")
plt.xticks(np.arange(N))
plt.grid(True)
plt.tight_layout()
plt.show()
```





2.2

```
import numpy as np
import matplotlib.pyplot as plt

x = np.array([2,4,6,8,10,8,6,4], dtype=complex)
N = len(x)
print("Sinyal input:", x.real)
print("Jumlah sampel:", N)

jumlah_bit = int(np.log2(N))

indeks = [
    int(format(i, f'0{jumlah_bit}b')[::-1], 2)
    for i in range(N)
]
data = x[indeks]

print("\nUrutan bit reversal:", indeks)
print("Data setelah bit reversal:", data.real)
```

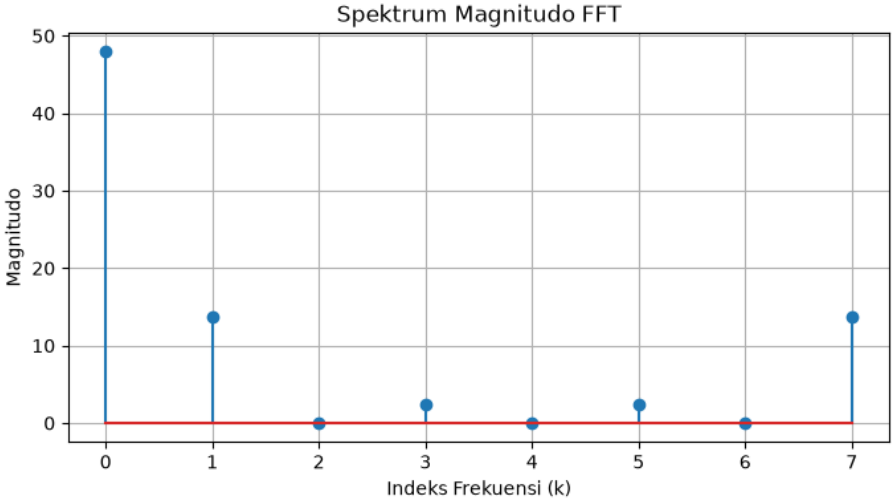
```
hasil = data.copy()
stage = 1
m = 2
while m <= N:
    half = m // 2
    for k in range(0, N, m):
        for j in range(half):
            W = np.exp(-2j * np.pi * j / m)
            a = hasil[k + j]
            b = hasil[k + j + half] * W
            hasil[k + j] = a + b
            hasil[k + j + half] = a - b
    print(f"\nHasil Stage {stage}:")
    print(np.round(hasil, 4))
    stage += 1
    m *= 2
```

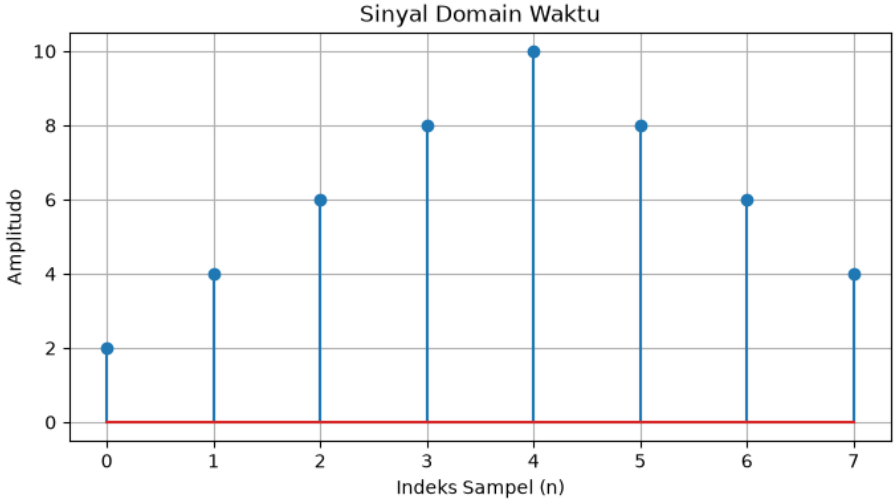
```
print("\nHASIL AKHIR FFT")
for k in range(N):
    print(f"X[{k}] = {hasil[k]:.4f}")

print("\nX[0] =", hasil[0].real)
print("X[1] =", hasil[1].real)
```

```
n = np.arange(N)
plt.figure(figsize=(8,4))
plt.stem(n, x.real)
plt.title("Sinyal Domain Waktu")
plt.xlabel("Indeks Sampel (n)")
plt.ylabel("Amplitudo")
plt.grid()
plt.show()
```

```
plt.figure(figsize=(8,4))
plt.stem(np.arange(N), np.abs(hasil))
plt.title("Spektrum Magnitudo FFT")
plt.xlabel("Indeks Frekuensi (k)")
plt.ylabel("Magnitudo")
plt.grid()
plt.show()
```





Butterfly diagram FFT 8 titik (radix-2 DIT), W8^k = e^(-j2*pi*k/8)

