

ayx7wy98a

October 9, 2024

## 0.1 Hands On 1

### 0.1.1 Topics

*Working With Signal*

### 0.1.2 Personal Identity

Nama Mahasiswa	Nomor Induk Mahasiswa
Kevin Simorangkir	121140150

### 0.1.3 Instructions

#### 1. Membuat Gelombang Sinus dan Cosinus:

- Buat array  $t$  yang berkisar dari 0 hingga 2 dengan langkah sebesar 0.0001.
- Buat sinyal-sinyal berikut menggunakan array waktu ini:
  1.  $y1 = 2 \cdot \sin(2\pi \cdot 3 \cdot t + 0)$
  2.  $y2 = 1 \cdot \cos(2\pi \cdot 4 \cdot t + \pi/4)$
  3.  $y3 = -1 \cdot \sin(2\pi \cdot 5 \cdot t + \pi/2)$
  4.  $y4 = 0.5 \cdot \cos(2\pi \cdot 6 \cdot t + \pi)$

#### 2. Perbandingan Subplot:

- Buat gambar dengan 4 subplot (grid 2x2) untuk membandingkan semua sinyal secara berdampingan.
- Setiap subplot harus berisi salah satu sinyal ( $y1$ ), ( $y2$ ), ( $y3$ ), dan ( $y4$ ).

#### 3. Pertanyaan Analisis:

- Jawab pertanyaan berikut dalam sel markdown di Jupyter Notebook Anda:
  1. Berapa amplitudo dan frekuensi masing-masing sinyal?
  2. Bagaimana pergeseran fase mempengaruhi posisi gelombang? (Anda dapat mengubah nilai fase pada sinyal-sinyal yang telah dibuat sesuka anda)
  3. Bandingkan sinyal-sinyal dengan amplitudo yang berbeda dan diskusikan bagaimana amplitudo mempengaruhi tampilan gelombang.
  4. Bandingkan sinyal-sinyal dengan pergeseran fase yang berbeda dan diskusikan bagaimana pergeseran fase mempengaruhi tampilan gelombang.

#### 4. Tugas Lanjutan:

- Buat sinyal baru  $y5$  yang merupakan kombinasi dari  $y1$  dan  $y2$  yaitu,  $y5 = y1 + y2$ .

- Plot  $y_5$  dan diskusikan bagaimana kombinasi dua gelombang sinus/cosinus mempengaruhi bentuk gelombang yang dihasilkan.
5. Buktikanlah bahwa proses downsampling (resampling dengan laju sampling yang lebih rendah) dapat menghilangkan informasi dari sinyal asli. Untuk melakukan hal ini, gunakan sinyal ECG sintetis (dengan method `nk.ecg_simulate`) sesuai spesifikasi berikut:
    - Durasi: Berdasarkan 3 digit terakhir nim anda
    - Sampling Rate: 150 Hz
    - Noise Level:  $0.<2 \text{ digit nim terakhir}>$
    - Heart Rate: 80 BPM
    - Random State: tanggal bulan tahun lahir anda dengan format YYMMDD misalnya 240925

Lakukan downsampling dari 150Hz ke 100Hz, 50Hz, 25Hz, 10Hz, hingga 5Hz. Jelaskan apa yang terjadi dan buktikan bahwa semakin rendah sampling frequency (fs) maka sinyal akan semakin terdistorsi dan terdapat Aliasing pada sinyal hasil downsampling. Jelaskan apa itu Aliasing

6. Terdapat parameter order pada saat melakukan filtering. Apa yang dimaksud dengan order? Apa yang terjadi ketika mengubah nilai order? Lakukan eksperimen secara mandiri
7. Lakukan eksperimen dengan merancang filter band-pass menggunakan `signal.butter`. Pada sinyal respirasi (pernapasan). Anda dapat dengan bebas menentukan sinyal asli, noise, dan frekuensi cutoff yang diinginkan. Jelaskan latar belakang penentuan frekuensi cutoff

#### 0.1.4 What's The Problem ?

##### 1. Membuat Gelombang Sinus dan Cosinus:

- Buat array  $t$  yang berkisar dari 0 hingga 2 dengan langkah sebesar 0.0001.
- Buat sinyal-sinyal berikut menggunakan array waktu ini:
  1.  $y_1 = 2 \cdot \sin(2\pi \cdot 3 \cdot t + 0)$
  2.  $y_2 = 1 \cdot \cos(2\pi \cdot 4 \cdot t + \pi/4)$
  3.  $y_3 = -1 \cdot \sin(2\pi \cdot 5 \cdot t + \pi/2)$
  4.  $y_4 = 0.5 \cdot \cos(2\pi \cdot 6 \cdot t + \pi)$

#### 0.1.5 Tahap Persiapan

1. Siapkan Library / pustaka yang menjadi alat dalam pengerjaan Hands On 1 ini. Dalam hal ini menggunakan 2 (dua) pustaka yaitu Numpy dan Matplotlib.pyplot.

```
[77]: import numpy as np # for numerical operations
import matplotlib.pyplot as plt # for plotting
```

2. Deklarasikan untuk sumbu waktu / sumbu x nya dalam array

```
[78]: sumbu_waktu = np.arange(0, 10, 0.0001) # sumbu waktu dari 0 sampai 10 detik
      ↪ dengan step 0.0001 detik
```

3. Definisikan sinyal dari  $y_1$  ,  $y_2$  ,  $y_3$ , dan  $y_4$  .

```
[79]: y1 = 2 * np.sin(2 * np.pi * 3 * sumbu_waktu + 0) # sinyal pertama
      y2 = 1 * np.cos(2 * np.pi * 4 * sumbu_waktu + np.pi/4) # sinyal kedua
      y3 = -1 * np.sin(2 * np.pi * 5 * sumbu_waktu + np.pi/2) # sinyal ketiga
      y4 = 0.5 * np.cos(2 * np.pi * 6 * sumbu_waktu + np.pi) # sinyal keempat
```

4. Visualisasikan hasil dari sinyal yang telah didefinisikan sebelumnya menjadi sebuah visual gambar.

```
[80]: plt.figure(figsize=(12, 8)) # ukuran plot

# Plot y1
plt.subplot(2, 2, 1)
plt.plot(sumbu_waktu, y1, label='Sinyal Sinusoidal y1')
plt.title('Sinyal')
plt.xlabel('Waktu (s)')
plt.ylabel('Amplitudo')
plt.grid()
plt.legend()

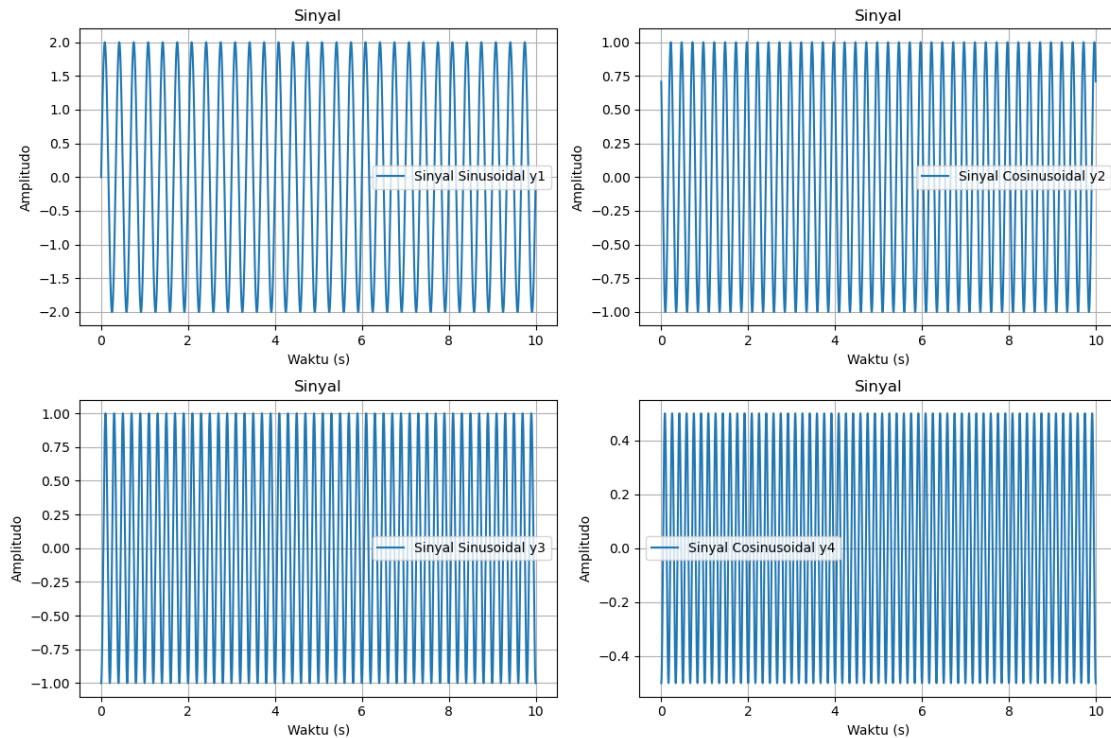
# Plot y2
plt.subplot(2, 2, 2)
plt.plot(sumbu_waktu, y2, label='Sinyal Cosinusoidal y2')
plt.title('Sinyal')
plt.xlabel('Waktu (s)')
plt.ylabel('Amplitudo')
plt.grid()
plt.legend()

# Plot y3
plt.subplot(2, 2, 3)
plt.plot(sumbu_waktu, y3, label='Sinyal Sinusoidal y3')
plt.title('Sinyal')
plt.xlabel('Waktu (s)')
plt.ylabel('Amplitudo')
plt.grid()
plt.legend()

# Plot y4
plt.subplot(2, 2, 4)
plt.plot(sumbu_waktu, y4, label='Sinyal Cosinusoidal y4')
plt.title('Sinyal')
plt.xlabel('Waktu (s)')
plt.ylabel('Amplitudo')
plt.grid()
plt.legend()

plt.tight_layout()
```

```
plt.show()
```



### 0.1.6 Mari Kita Analisis (MaKiAn)

#### 1. Amplitudo dan Frekuensi Masing-Masing Sinyal:

- Sinyal ( y1 ):
  - Amplitudo: 2
  - Frekuensi: 3 Hz
- Sinyal ( y2 ):
  - Amplitudo: 1
  - Frekuensi: 4 Hz
- Sinyal ( y3 ):
  - Amplitudo: 1
  - Frekuensi: 5 Hz
- Sinyal ( y4 ):
  - Amplitudo: 0.5
  - Frekuensi: 6 Hz

#### 2. Pengaruh Pergeseran Fase Terhadap Posisi Gelombang:

- Pergeseran fase menggeser posisi gelombang sepanjang sumbu waktu. Misalnya, pada sinyal ( y2 ), pergeseran fase sebesar (  $\pi/4$  ) menyebabkan gelombang bergeser ke kiri atau ke kanan tergantung pada tanda pergeseran fase tersebut. Pergeseran fase tidak mengubah amplitudo atau frekuensi gelombang, hanya posisi puncak dan lembah gelombang yang berubah.

#### 3. Pengaruh Amplitudo Terhadap Tampilan Gelombang:

- Amplitudo menentukan tinggi puncak dan rendah lembah dari gelombang. Sinyal dengan amplitudo yang lebih besar akan memiliki puncak yang lebih tinggi dan lembah yang lebih dalam. Misalnya, sinyal (  $y_1$  ) dengan amplitudo 2 memiliki puncak yang lebih tinggi dibandingkan dengan sinyal (  $y_4$  ) yang memiliki amplitudo 0.5.
4. **Pengaruh Pergeseran Fase Terhadap Tampilan Gelombang:**
- Pergeseran fase mengubah posisi gelombang sepanjang sumbu waktu tanpa mengubah bentuk dasar gelombang. Misalnya, sinyal (  $y_3$  ) dengan pergeseran fase (  $\pi/2$  ) akan memiliki puncak dan lembah yang bergeser dibandingkan dengan sinyal tanpa pergeseran fase. Pergeseran fase dapat menyebabkan dua gelombang dengan frekuensi dan amplitudo yang sama terlihat berbeda dalam hal posisi relatif puncak dan lembahnya.

### 0.1.7 What's The Problem?

#### 2. Perbandingan Subplot:

- Buat gambar dengan 4 subplot (grid 2x2) untuk membandingkan semua sinyal secara berdampingan.
- Setiap subplot harus berisi salah satu sinyal (  $y_1$  ), (  $y_2$  ), (  $y_3$  ), dan (  $y_4$  ).

```
[81]: fig, ax = plt.subplots(2, 2, figsize=(12, 10))

ax[0, 0].plot(sumbu_waktu, y1, label='Sinyal Sinusoidal y1')
ax[0, 0].set_title('Sinyal y1')
ax[0, 0].set_xlabel('Waktu (s)')
ax[0, 0].set_ylabel('Amplitudo')
ax[0, 0].grid()
ax[0, 0].legend()

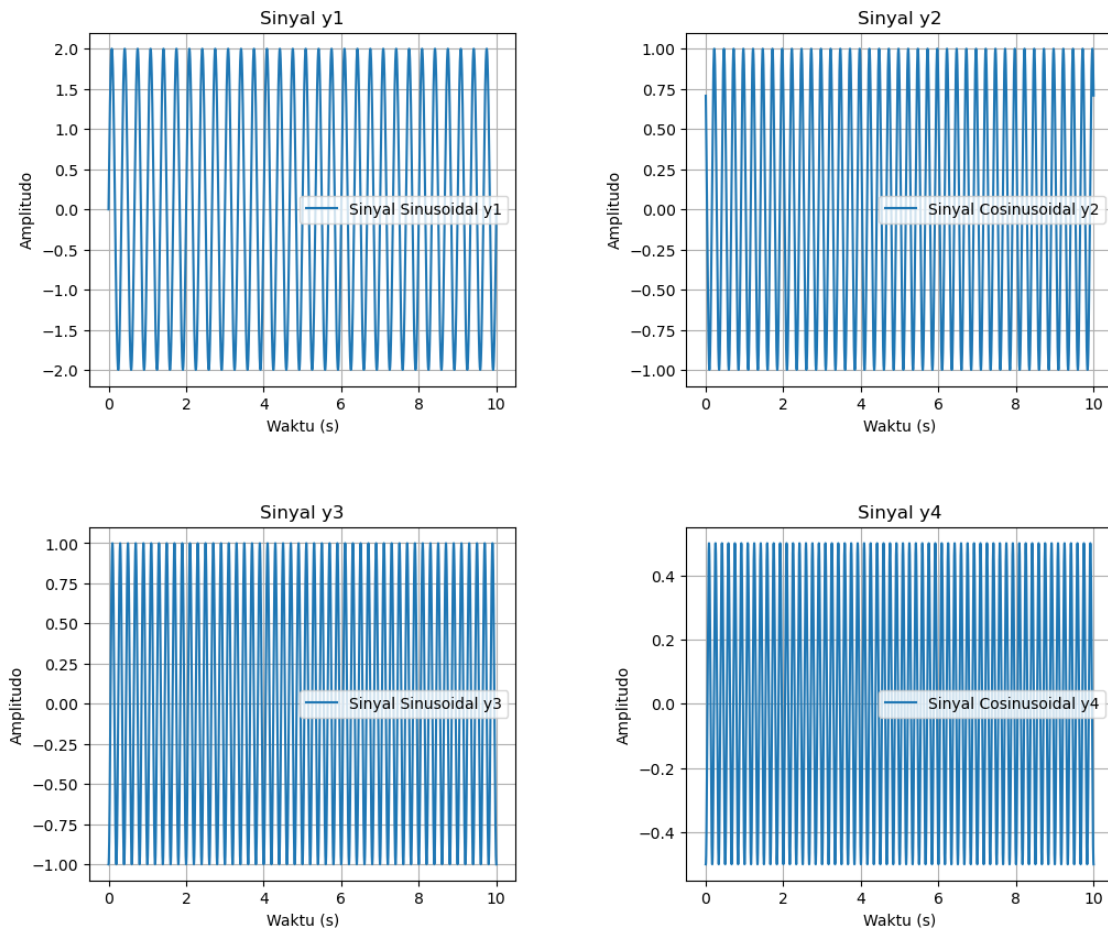
ax[0, 1].plot(sumbu_waktu, y2, label='Sinyal Cosinusoidal y2')
ax[0, 1].set_title('Sinyal y2')
ax[0, 1].set_xlabel('Waktu (s)')
ax[0, 1].set_ylabel('Amplitudo')
ax[0, 1].grid()
ax[0, 1].legend()

ax[1, 0].plot(sumbu_waktu, y3, label='Sinyal Sinusoidal y3')
ax[1, 0].set_title('Sinyal y3')
ax[1, 0].set_xlabel('Waktu (s)')
ax[1, 0].set_ylabel('Amplitudo')
ax[1, 0].grid()
ax[1, 0].legend()

ax[1, 1].plot(sumbu_waktu, y4, label='Sinyal Cosinusoidal y4')
ax[1, 1].set_title('Sinyal y4')
ax[1, 1].set_xlabel('Waktu (s)')
ax[1, 1].set_ylabel('Amplitudo')
ax[1, 1].grid()
```

```
ax[1, 1].legend()

plt.subplots_adjust(hspace=0.4, wspace=0.4)
plt.show()
plt.close()
```



### 0.1.8 What's The Problem

#### 3. Pertanyaan Analisis:

- Jawab pertanyaan berikut dalam sel markdown di Jupyter Notebook Anda:
  - Berapa amplitudo dan frekuensi masing-masing sinyal?
  - Bagaimana pergeseran fase mempengaruhi posisi gelombang? (Anda dapat mengubah nilai fase pada sinyal-sinyal yang telah dibuat sesuai keinginan anda)
  - Bandingkan sinyal-sinyal dengan amplitudo yang berbeda dan diskusikan bagaimana amplitudo mempengaruhi tampilan gelombang.
  - Bandingkan sinyal-sinyal dengan pergeseran fase yang berbeda dan diskusikan bagaimana pergeseran fase mempengaruhi tampilan gelombang.

Soal	Jawaban
Berapa amplitudo dan frekuensi masing-masing sinyal?	Amplitudo dari $y_1$ adalah 2, frekuensi adalah 3 Hz. ; Amplitudo dari $y_2$ adalah 1, frekuensi adalah 4 Hz. ; Amplitudo dari $y_3$ adalah 1, frekuensi adalah 5 Hz. ; Amplitudo dari $y_4$
Bagaimana pergeseran fase mempengaruhi posisi gelombang? (Anda dapat mengubah nilai fase pada sinyal-sinyal yang telah dibuat sesuka anda)	Pergeseran fase menggeser posisi gelombang secara horizontal. Misalnya, pada $y_2$ , fase $\pi/4$ menggeser gelombang ke kanan, sementara pada $y_3$ , fase $\pi/2$ menggeser gelombang lebih jauh ke kanan.
Bandingkan sinyal-sinyal dengan amplitudo yang berbeda dan diskusikan bagaimana amplitudo mempengaruhi tampilan gelombang	Sinyal dengan amplitudo lebih besar memiliki puncak dan lembah yang lebih tinggi. Sebagai contoh, $y_1$ dengan amplitudo 2 terlihat lebih tinggi dibandingkan $y_4$ yang hanya memiliki amplitudo 0.5.
Bandingkan sinyal-sinyal dengan pergeseran fase yang berbeda dan diskusikan bagaimana pergeseran fase mempengaruhi tampilan gelombang.	Pergeseran fase menggeser titik awal gelombang pada sumbu waktu. Semakin besar pergeseran fase, semakin banyak pergeseran gelombang secara horizontal (misalnya, $y_3$ dengan fase $\pi/2$ dibandingkan dengan $y_1$ yang tidak memiliki pergeseran fase).

## Jawaban

### 0.1.9 What's The Problem?

#### 4. Tugas Lanjutan:

- Buat sinyal baru  $y_5$  yang merupakan kombinasi dari  $y_1$  dan  $y_2$  yaitu,  $y_5 = y_1 + y_2$ .
- Plot  $y_5$  dan diskusikan bagaimana kombinasi dua gelombang sinus/cosinus mempengaruhi bentuk gelombang yang dihasilkan.

### 0.1.10 Tahap Eksekusi

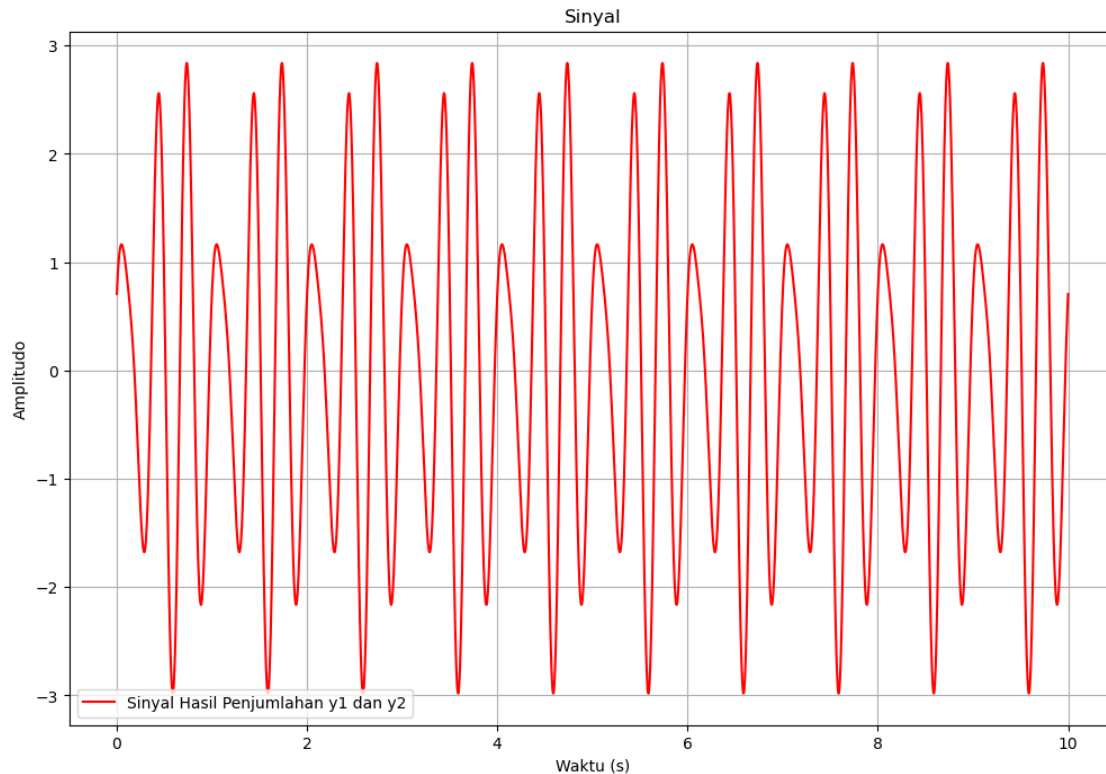
1. Deklarasikan sinyal baru yang dalam hal ini  $y_5$  dengan kombinasi yang sudah ditentukan dalam What's The Problem

```
[82]: y5 = y1 + y2 #hasil penjumlahan sinyal y1 dan y2
```

2. Visualisasikan gambar sinyal baru yang telah ditambahkan dalam kombinasi ini

```
[83]: plt.figure(figsize=(12, 8))
plt.plot(sumbu_waktu, y5, label='Sinyal Hasil Penjumlahan y1 dan y2',
        color='red')
plt.title('Sinyal')
plt.xlabel('Waktu (s)')
```

```
plt.ylabel('Amplitudo')
plt.grid()
plt.legend()
plt.show()
plt.close()
```



### 0.1.11 Mari Kita Analisis (MaKiAn)

Sinyal  $y_5$  adalah hasil penjumlahan dari dua gelombang sinusoidal dengan frekuensi dan amplitudo yang berbeda. Kombinasi ini menghasilkan interferensi yang menciptakan pola gelombang baru dengan bentuk yang lebih kompleks, tergantung pada fase dan amplitudo masing-masing sinyal.

### 0.1.12 What's The Problem

Buktikanlah bahwa proses downsampling (resampling dengan laju sampling yang lebih rendah) dapat menghilangkan informasi dari sinyal asli. Untuk melakukan hal ini, gunakan sinyal ECG sintetis (dengan method `nk.ecg_simulate`) sesuai spesifikasi berikut:

- Durasi: Berdasarkan 3 digit terakhir nim anda
- Sampling Rate: 150 Hz
- Noise Level:  $0.<2 \text{ digit nim terakhir}>$
- Heart Rate: 80 BPM



- Random State: tanggal bulan tahun lahir anda dengan format YYMMDD misalnya 240925

Lakukan downsampling dari 150Hz ke 100Hz, 50Hz, 25Hz, 10Hz, hingga 5Hz. Jelaskan apa yang terjadi dan buktikan bahwa semakin rendah sampling frequency (fs) maka sinyal akan semakin terdistorsi dan terdapat Aliasing pada sinyal hasil downsampling. Jelaskan apa itu Aliasing

### 0.1.13 Tahap Persiapan

1. Siapkan Library / pustaka yang menjadi alat dalam pengerjaan Hands On 1 ini. Dalam hal ini menggunakan 2 (dua) pustaka yaitu Numpy dan Matplotlib.pyplot, neurokit2 dan pywt

```
[84]: import numpy as np
import matplotlib.pyplot as plt
import neurokit2 as nk
```

2. Deklarasikan yang sesuai dengan spesifikasi yang diberikan yaitu sebagai berikut :
  - Durasi: Berdasarkan 3 digit terakhir nim anda
  - Sampling Rate: 150 Hz
  - Noise Level: 0.<2 digit nim terakhir>
  - Heart Rate: 80 BPM
  - Random State: tanggal bulan tahun lahir anda dengan format YYMMDD misalnya 240925

```
[85]: rate_sample = 150 # frekuensi sampling 150 Hz
durasi = 150 # durasi 5 detik
level_noise = 0.50 # level noise 0.50
rate_heart = 80 # frekuensi jantung 80 BPM
state_random = 140803 # seed random
```

3. Deklarasikan sinyal EEG nya

```
[86]: signal_ecg = nk.ecg_simulate(duration=durasi, heart_rate=rate_heart,
↳ noise=level_noise, random_state=state_random)
```

4. Buatlah Downsampling ke berbagai laju sampling

```
[87]: rate_sample = [100, 50, 25, 10, 5] # frekuensi sampling 100, 50, 25, 10, 5 Hz
signals_downsampled = {} # inisialisasi dictionary
```

5. Buatlah Visualisasi dari hasil yang telah ditemukan dari rangkaian cara

```
[88]: # Perform downsampling
for fs in rate_sample:
    downsampled_signal = signal_ecg[:,int(150/fs)]
    signals_downsampled[fs] = downsampled_signal

# Plot the original and downsampled signals
plt.figure(figsize=(12, 12))
plt.subplot(3, 2, 1)
```

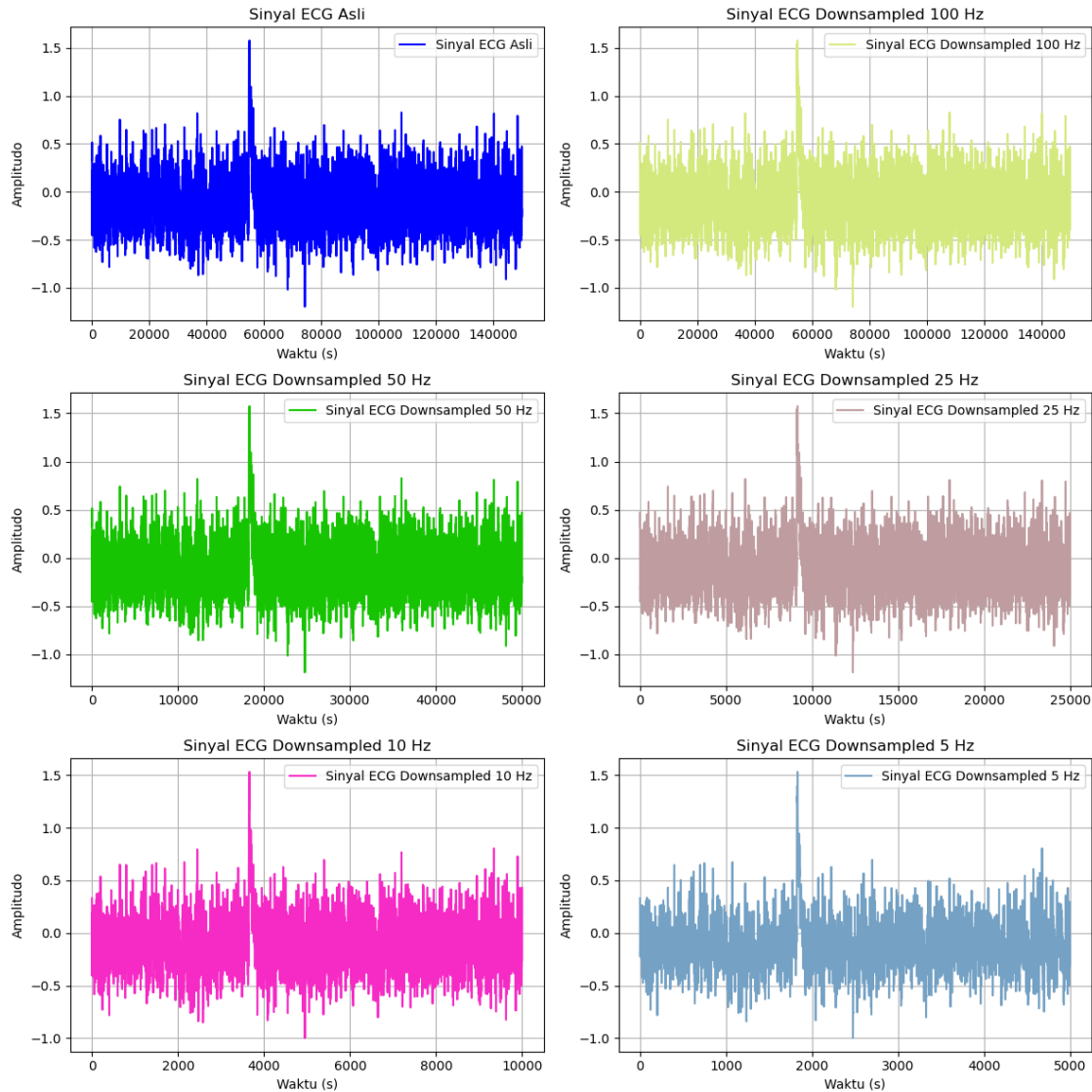
```

plt.plot(signal_ecg, label='Sinyal ECG Asli', color='blue')
plt.title('Sinyal ECG Asli')
plt.xlabel('Waktu (s)')
plt.ylabel('Amplitudo')
plt.legend()
plt.grid()

for i, fs in enumerate(rate_sample):
    plt.subplot(3, 2, i+2)
    plt.plot(signals_downsampled[fs], label='Sinyal ECG Downsampled {} Hz'.
    ↳format(fs), color=np.random.rand(3,)) # plot sinyal downsampled dengan warna
    ↳acak
    plt.title('Sinyal ECG Downsampled {} Hz'.format(fs))
    plt.xlabel('Waktu (s)')
    plt.ylabel('Amplitudo')
    plt.legend()
    plt.grid()

plt.tight_layout()
plt.show()
plt.close()

```



#### 0.1.14 Mari Kita Analisis (MaKiAn)

1. Jelaskan apa yang terjadi dan buktikan bahwa semakin rendah sampling frequency ( $f_s$ ) maka sinyal akan semakin terdistorsi dan terdapat Aliasing pada sinyal hasil downsampling!

**Jawaban :**

Sinyal Asli (150 Hz): Menampilkan bentuk gelombang ECG yang jelas, mencakup semua detail. Downsampling: Saat kita menurunkan laju sampling (ke 100 Hz, 50 Hz, 25 Hz, 10 Hz, dan 5 Hz), kita mulai kehilangan detail dari sinyal asli. 100 Hz: Masih cukup jelas, tetapi beberapa detail mulai hilang. 50 Hz: Terlihat distorsi lebih signifikan. 25 Hz: Lebih banyak detail yang hilang; bentuk gelombang mulai menyusut. 10 Hz: Banyak informasi yang hilang; sinyal terlihat lebih kasar. 5 Hz: Hampir tidak ada detail ECG yang tersisa; hanya beberapa puncak yang terlihat.

2. Jelaskan apa itu Aliasing!

**Jawaban :**

fenomena yang terjadi ketika sinyal dengan frekuensi tinggi (komponen frekuensi tinggi) tidak dapat terwakili dengan baik oleh laju sampling yang rendah

---

#### 0.1.15 What's The Problem

6. Terdapat parameter order pada saat melakukan filtering. Apa yang dimaksud dengan order? Apa yang terjadi ketika mengubah nilai order? Lakukan eksperimen secara mandiri

#### 0.1.16 Mari Kita Analisis (MaKiAn)

1. Apa yang dimaksud dengan order?

**Jawaban :**

Order filter adalah jumlah elemen atau derajat dalam fungsi transfer filter. Sebuah filter dengan order lebih tinggi akan memiliki lebih banyak koefisien dalam desainnya, yang berarti dapat memproduksi respons frekuensi yang lebih tajam dan lebih terfokus

2. Apa yang terjadi ketika mengubah nilai order?

**Jawaban :**

Mengubah nilai order filter akan mempengaruhi karakteristik filter tersebut. Filter dengan order yang lebih tinggi akan memiliki transisi yang lebih tajam antara passband dan stopband, yang berarti filter tersebut lebih efektif dalam memisahkan frekuensi yang diinginkan dari frekuensi yang tidak diinginkan. Namun, filter dengan order yang lebih tinggi juga dapat menyebabkan lebih banyak distorsi fase dan membutuhkan lebih banyak komputasi. Sebaliknya, filter dengan order yang lebih rendah akan memiliki transisi yang lebih lembut dan mungkin tidak seefektif dalam memisahkan frekuensi, tetapi akan menyebabkan lebih sedikit distorsi fase dan membutuhkan lebih sedikit komputasi.

#### 0.1.17 Tahap Eksekusi

1. Siapkan Library / pustaka yang menjadi alat dalam pengerjaan Hands On 1 ini. Dalam hal ini menggunakan 2 (dua) pustaka yaitu Numpy dan Matplotlib.pyplot, neurokit2 dan butter .

```
[89]: import numpy as np
import matplotlib.pyplot as plt
import neurokit2 as nk
from scipy.signal import butter, filtfilt
```

2. Deklarasikan yang sesuai dengan spesifikasi yang diberikan yaitu sebagai berikut :
  - Durasi: Berdasarkan 3 digit terakhir nim anda
  - Sampling Rate: 150 Hz
  - Noise Level: 0.<2 digit nim terakhir>
  - Heart Rate: 80 BPM
  - Random State: tanggal bulan tahun lahir anda dengan format YYMMDD misalnya 240925

```
[90]: rate_sample = 150 # frekuensi sampling 150 Hz
      durasi = 150 # durasi 5 detik
      level_noise = 0.50 # level noise 0.50
      rate_heart = 80 # frekuensi jantung 80 BPM
      state_random = 140803 # seed random
```

3. Deklarasikan sinyal EEG nya

```
[91]: signal_ecg = nk.ecg_simulate(
      duration=durasi, heart_rate=rate_heart, noise=level_noise,
      ↪random_state=state_random)
```

4. Deklarasikan untuk fungsi membuat dan menerapkan filter Butterworth

```
[92]: def lowpass_butter (cutoff, fs, order=5):
      nyquist = 0.5 * fs
      normal_cutoff = cutoff / nyquist
      b, a = signal.butter(order, normal_cutoff, btype='low', analog=False)
      return b, a

      def lowpass_filter (data, cutoff, fs, order=5):
          b, a = lowpass_butter(cutoff, fs, order=order)
          y = signal.filtfilt(b, a, data)
          return y
```

5. Deklarasi filter sinyal dengan berbagai order

```
[93]: freq_cutoff = 10 # frekuensi cutoff 10 Hz
      orders = [1, 3, 5, 8] # orde filter 1, 3, 5, 8
```

```
[94]: from scipy import signal

      filtered_signals = {} # inisialisasi dictionary
      for order in orders:
          filtered_signal = lowpass_filter(signal_ecg, freq_cutoff, rate_sample,
          ↪order=order)
          filtered_signals[order] = filtered_signal
```

6. Tampilkan visualisasi nya dengan tampilan yang sesuai dari hasil yang telah didetailkan

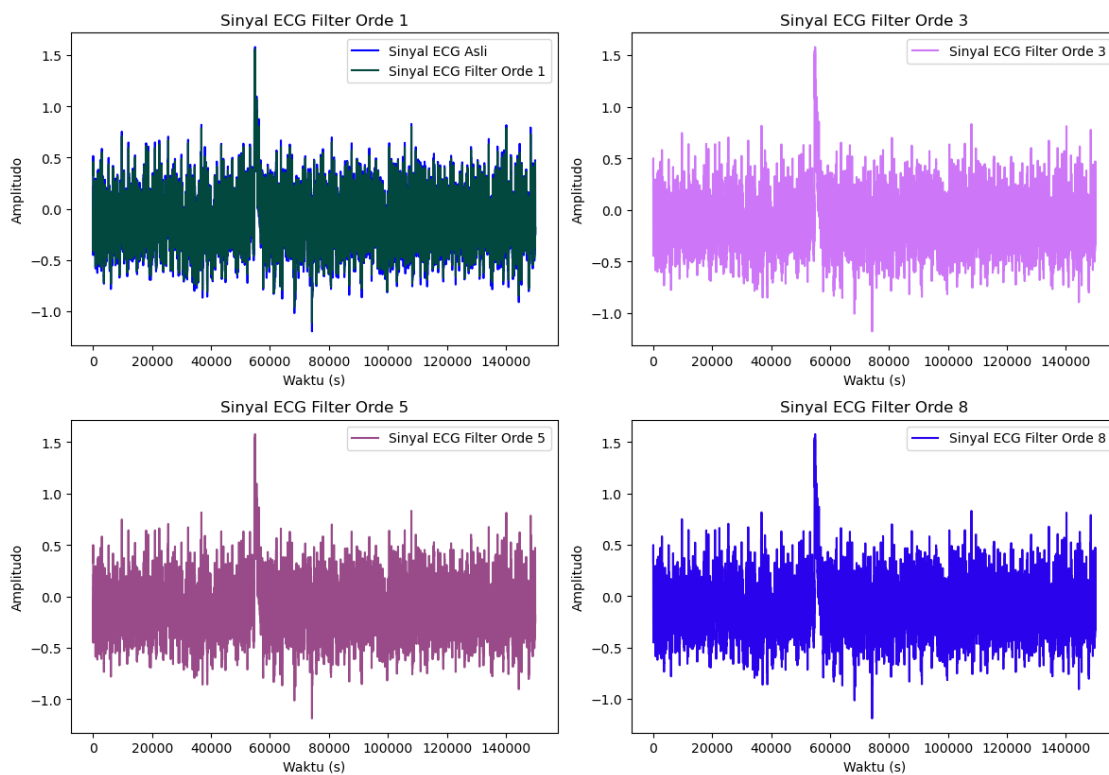
```
[95]: plt.figure(figsize=(12, 12))
      plt.subplot(3, 2, 1)
      plt.plot(signal_ecg, label='Sinyal ECG Asli', color='blue')
      plt.title('Sinyal ECG Asli')
      plt.xlabel('Waktu (s)')
      plt.ylabel('Amplitudo')
      plt.legend()
```

```

for i, order in enumerate(orders):
    plt.subplot(3, 2, i+1)
    plt.plot(filtered_signals[order], label='Sinyal ECG Filter Orde {}'.
    ↪format(order), color=np.random.rand(3,))
    plt.title('Sinyal ECG Filter Orde {}'.format(order))
    plt.xlabel('Waktu (s)')
    plt.ylabel('Amplitudo')
    plt.legend()

plt.tight_layout()
plt.show()
plt.close()

```



### 0.1.18 What's The Problem

7. Lakukan eksperimen dengan merancang filter band-pass menggunakan `signal.butter`. Pada sinyal respirasi (pernapasan). Anda dapat dengan bebas menentukan sinyal asli, noise, dan frekuensi cutoff yang diinginkan. Jelaskan latar belakang penentuan frekuensi cutoff

### 0.1.19 Tahap Eksekusi

1. Siapkan Library / pustaka yang menjadi alat dalam pengerjaan Hands On 1 ini. Dalam hal ini menggunakan 2 (dua) pustaka yaitu Numpy dan Matplotlib.pyplot, neurokit2 dan butter .

```
[96]: import numpy as np
import matplotlib.pyplot as plt
import neurokit2 as nk
from scipy.signal import butter, filtfilt
```

2. Deklarasikan dengan membuat sinyal respirasi sintesis

```
[97]: fs = 150 # frekuensi sampling 150 Hz
waktu_sumbu = np.linspace(0, 30, 30*fs) # sumbu waktu dari 0 sampai 10 detik

freq_respirasi = 0.2 # frekuensi pernapasan 0.2 Hz
signal_respirasi = 0.5 * np.sin(2 * np.pi * freq_respirasi * waktu_sumbu) #
↳sinyal pernapasan
```

3. Deklarasikan Noise Gaussian

```
[98]: level_noise = 0.1 # level noise 0.1
signal_noise = signal_respirasi + level_noise * np.random.
↳normal(size=len(waktu_sumbu)) # sinyal noise ditambahkan ke sinyal
↳pernapasan
```

4. Deklarasikan untuk fungsi membuat dan menerapkan filter Bandpass

```
[99]: def bandpass_butter(lowcut, highcut, fs, order=5):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = signal.butter(order, [low, high], btype='band', analog=False)
    return b, a

def filter_bandpass(data, lowcut, highcut, fs, order=5):
    b, a = bandpass_butter(lowcut, highcut, fs, order=order)
    y = signal.filtfilt(b, a, data)
    return y
```

5. Deklarasikan Cut Low dan High Cut

```
[100]: cutlow = 0.2 # frekuensi cut-off rendah 0.2 Hz
highcut = 0.3 # frekuensi cut-off tinggi 0.3 Hz
```

```
[101]: signal_filtered = filter_bandpass(signal_noise, cutlow, highcut, fs, order=5)
```

6. Deklarasikan Visualisasinya

```

[102]: plt.figure(figsize=(15, 10))

plt.subplot(3, 1, 1)
plt.plot(waktu_sumbu, signal_respirasi, label='Sinyal Respirasi Asli',
        color='green')
plt.title('Sinyal Respirasi Asli')
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitude')
plt.legend()

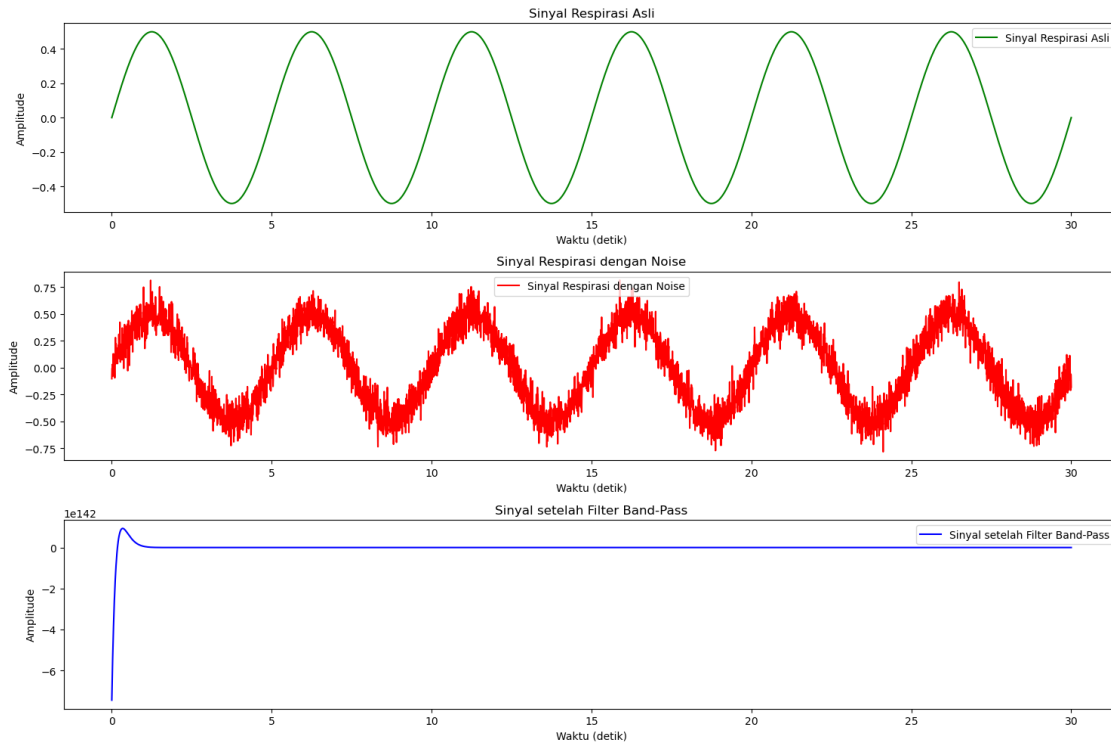
plt.subplot(3, 1, 2)
plt.plot(waktu_sumbu, signal_noise, label='Sinyal Respirasi dengan Noise',
        color='red')
plt.title('Sinyal Respirasi dengan Noise')
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitude')
plt.legend()

plt.subplot(3, 1, 3)
plt.plot(waktu_sumbu, signal_filtered, label='Sinyal setelah Filter Band-Pass',
        color='blue')
plt.title('Sinyal setelah Filter Band-Pass')
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitude')
plt.legend()

plt.tight_layout()
plt.show()

```





### 0.1.20 Mari Kita Analisis (MaKiAn)

**Soal :**

Jelaskan latar belakang penentuan frekuensi cutoff

**Jawaban :**

Frekuensi cutoff ditentukan berdasarkan karakteristik sinyal yang ingin dipertahankan dan noise yang ingin dihilangkan. Dalam kasus sinyal respirasi, frekuensi pernapasan manusia biasanya berada dalam rentang 0.1 hingga 0.5 Hz. Oleh karena itu, frekuensi cutoff rendah dipilih pada 0.2 Hz untuk memastikan bahwa komponen frekuensi rendah dari sinyal respirasi tetap dipertahankan. Frekuensi cutoff tinggi dipilih pada 0.3 Hz untuk menghilangkan noise frekuensi tinggi yang tidak diinginkan, sambil tetap mempertahankan komponen frekuensi utama dari sinyal respirasi. Pemilihan frekuensi cutoff ini memastikan bahwa sinyal yang dihasilkan setelah filter band-pass memiliki kualitas yang baik dan bebas dari noise yang mengganggu.

## 0.2 Libraries/Tools/Resources Used in the Work of Hands On 1

Nama Sumber/Alat	Tautan Sumber	Keterangan	Bukti
Chat GPT (1)	<a href="https://chatgpt.com/share/67069fa0-32b0-8001-b2c8-d1d8e78de2b2">https://chatgpt.com/share/67069fa0-32b0-8001-b2c8-d1d8e78de2b2</a>	Chapter 1 - 4 digunakan untuk membantu mencari solusi dari permasalahan yang diminta pada Number 1 - 4	-
Chat GPT (2)	<a href="https://chatgpt.com/share/67069b73-3f20-8001-9f5c-b3ae6715c0ed">https://chatgpt.com/share/67069b73-3f20-8001-9f5c-b3ae6715c0ed</a>	Chapter 5 - 7 digunakan untuk mencari solusi dari pertanyaan yang diberikan pada poin Number 5 - 7	-
2PDF	<a href="https://2pdf.com/converter/ipynb-to-pdf/">https://2pdf.com/converter/ipynb-to-pdf/</a>	2PDF digunakan untuk mengkonversi dari file .ipynb menjadi file .PDF	-