

Hands On 1

Instruksi Tugas

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:
 - A. $y1 = 2 \cdot \sin(2\pi \cdot 3 \cdot t + 0)$
 - B. $y2 = 1 \cdot \cos(2\pi \cdot 4 \cdot t + \pi/4)$
 - C. $y3 = -1 \cdot \sin(2\pi \cdot 5 \cdot t + \pi/2)$
 - D. $y4 = 0.5 \cdot \cos(2\pi \cdot 6 \cdot t + \pi)$

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:
 - A. Berapa amplitudo dan frekuensi masing-masing sinyal?
 - B. Bagaimana pergeseran fase mempengaruhi posisi gelombang? (Anda dapat mengubah nilai fase pada sinyal-sinyal yang telah dibuat sesuka anda)
 - C. Bandingkan sinyal-sinyal dengan amplitudo yang berbeda dan diskusikan bagaimana amplitudo mempengaruhi tampilan gelombang.
 - D. 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 $y5$ dan diskusikan bagaimana kombinasi dua gelombang sinus/cosinus mempengaruhi bentuk gelombang yang dihasilkan.

Identitas Diri

Nama

Nomor Induk Mahasiswa

Ghaza Muhammad Al Ghifari 121140215

Mata Kuliah

Menginstal Library

Instal library yang akan dibutuhkan dalam tugas ini, dalam kasus ini diperlukan hal-hal berikut:

- Library `numpy`
- Library `matplotlib`
- Library `OS`
- Library `wave`

Perintah Alat Library `!pip install numpy matplotlib wave`

Jawaban

1. Membuat Gelombang Sinus dan Cosinus:

```
In [99]: import numpy as np
import matplotlib.pyplot as plt

# Membuat array t dari 0 hingga 2 dengan langkah 0.0001
t = np.arange(0, 2, 0.0001)

# Membuat sinyal-sinyal
y1 = 2 * np.sin(2 * np.pi * 3 * t + 0)
y2 = 1 * np.cos(2 * np.pi * 4 * t + np.pi / 4)
y3 = -1 * np.sin(2 * np.pi * 5 * t + np.pi / 2)
y4 = 0.5 * np.cos(2 * np.pi * 6 * t + np.pi)

# Membuat plot untuk setiap sinyal
plt.figure(figsize=(10, 8))

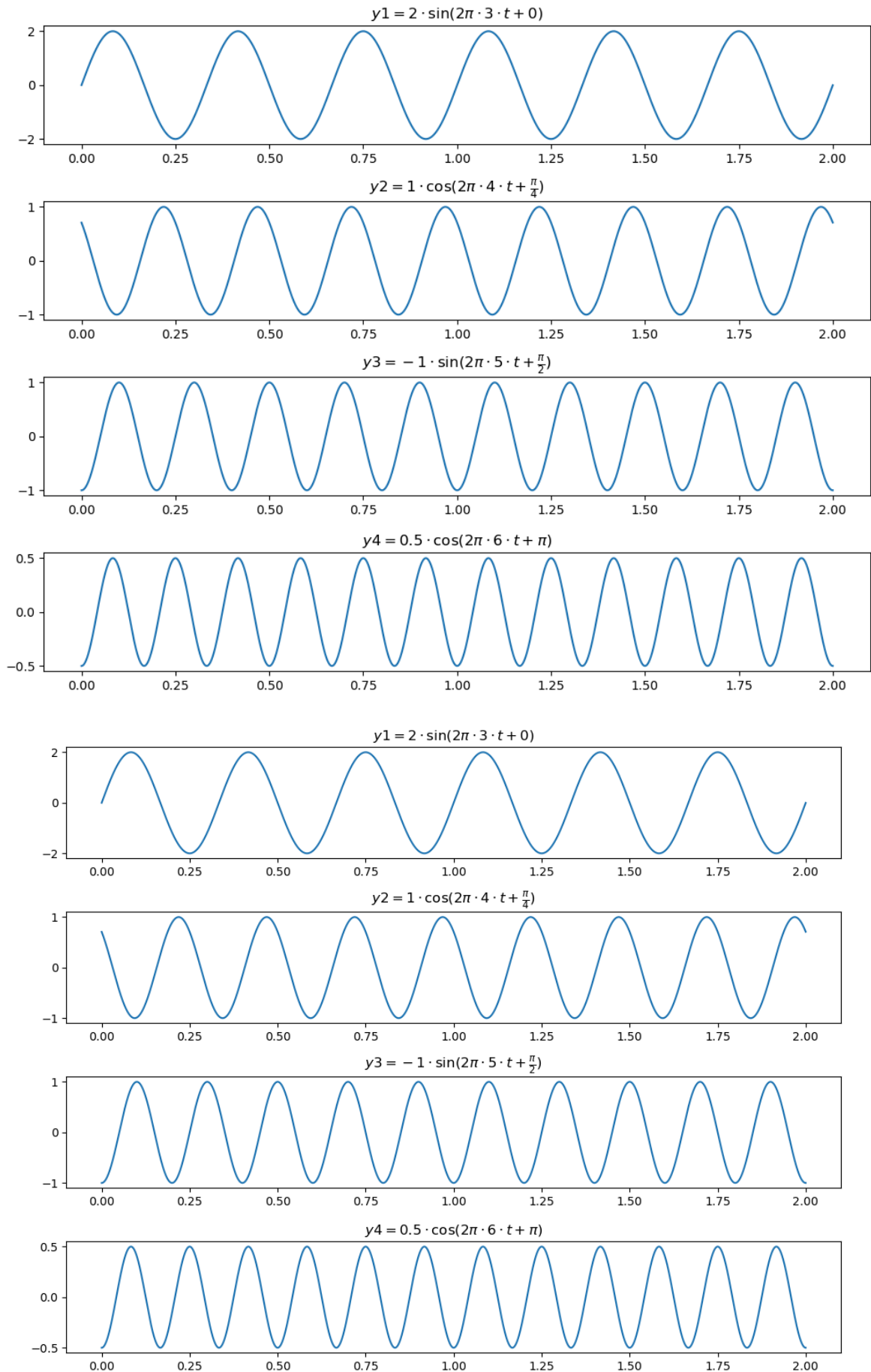
# Plot y1
plt.subplot(4, 1, 1)
plt.plot(t, y1)
plt.title(r'$y_1 = 2 \cdot \sin(2 \pi \cdot 3 \cdot t + 0)$')

# Plot y2
plt.subplot(4, 1, 2)
plt.plot(t, y2)
plt.title(r'$y_2 = 1 \cdot \cos(2 \pi \cdot 4 \cdot t + \frac{\pi}{4})$')

# Plot y3
plt.subplot(4, 1, 3)
plt.plot(t, y3)
plt.title(r'$y_3 = -1 \cdot \sin(2 \pi \cdot 5 \cdot t + \frac{\pi}{2})$')

# Plot y4
plt.subplot(4, 1, 4)
plt.plot(t, y4)
plt.title(r'$y_4 = 0.5 \cdot \cos(2 \pi \cdot 6 \cdot t + \pi)$')
```

```
# Mengatur layout agar tidak tumpang tindih  
plt.tight_layout()  
plt.show()
```



2. Perbandingan Subplot:

```
In [100... # Membuat gambar dengan 4 subplot (grid 2x2) untuk membandingkan semua si
plt.figure(figsize=(10, 8))

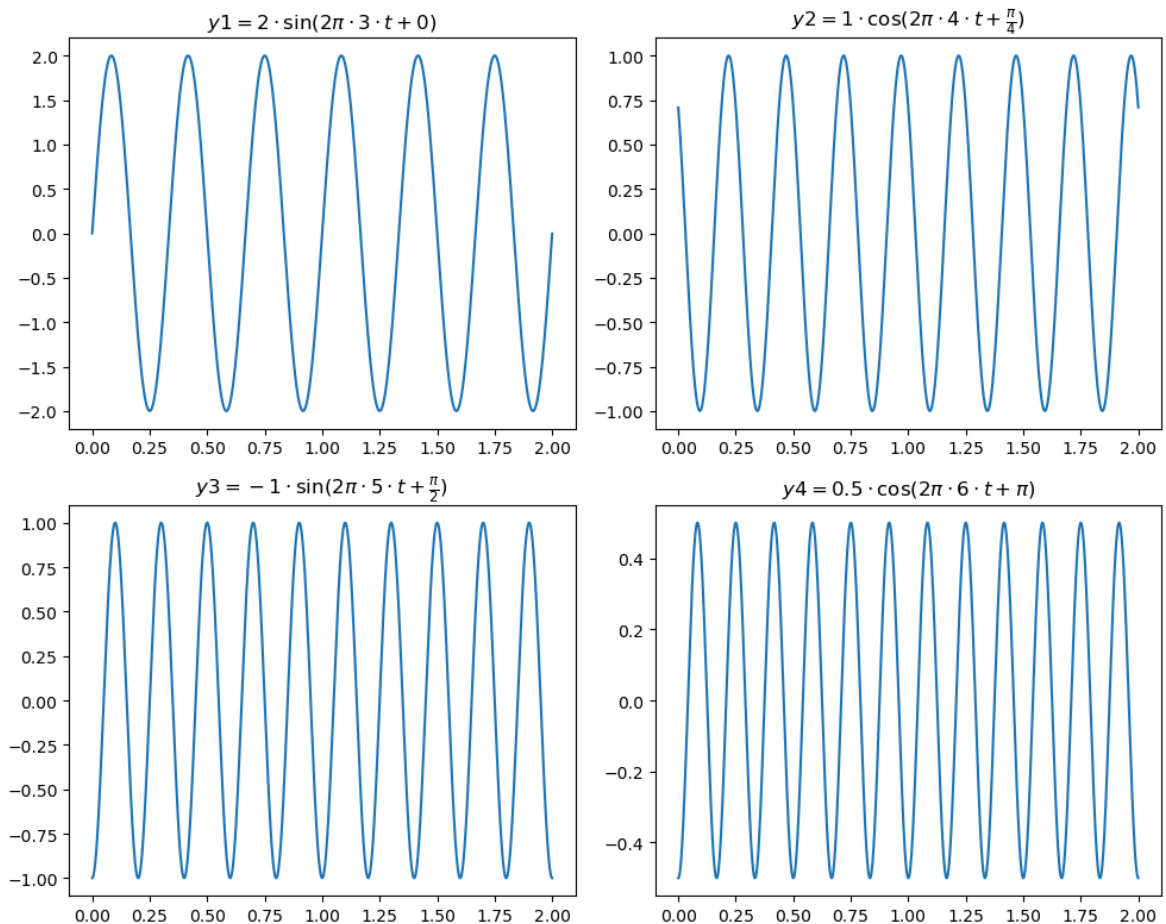
# Subplot 1: y1
plt.subplot(2, 2, 1)
plt.plot(t, y1)
plt.title(r'$y_1 = 2 \cdot \sin(2 \pi \cdot 3 \cdot t + 0)$')

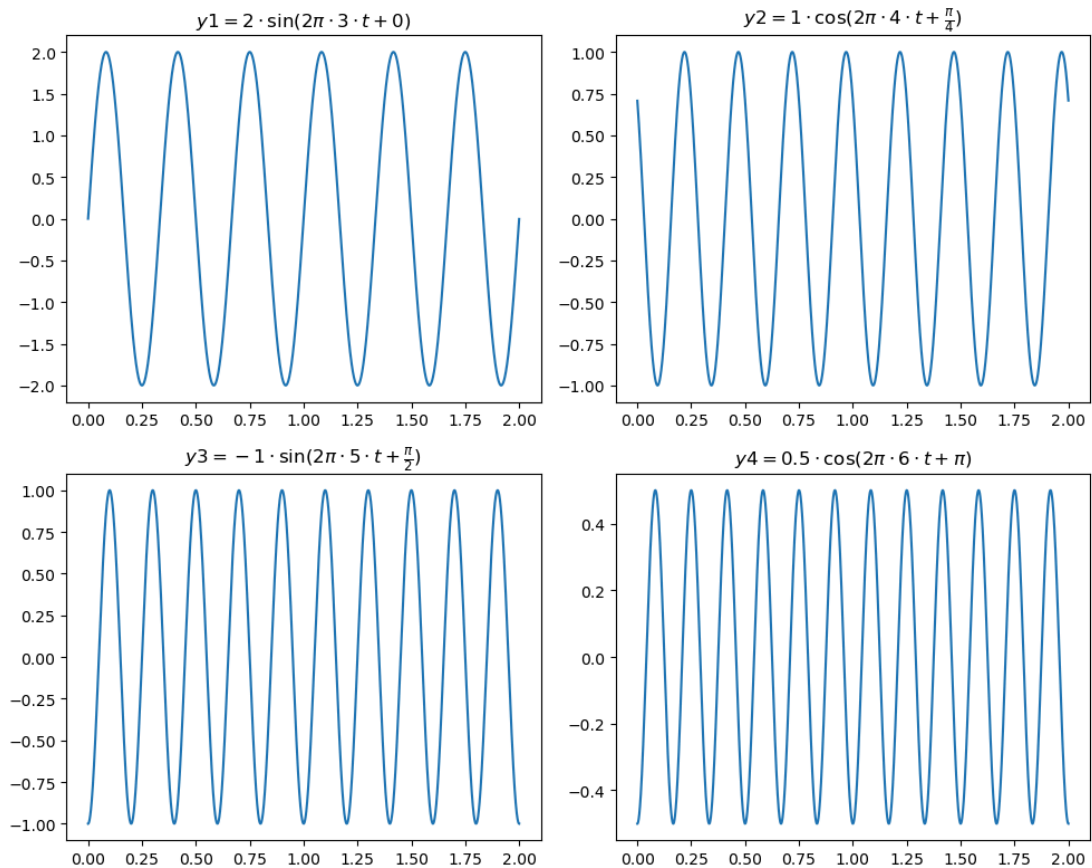
# Subplot 2: y2
plt.subplot(2, 2, 2)
plt.plot(t, y2)
plt.title(r'$y_2 = 1 \cdot \cos(2 \pi \cdot 4 \cdot t + \frac{\pi}{4})$')

# Subplot 3: y3
plt.subplot(2, 2, 3)
plt.plot(t, y3)
plt.title(r'$y_3 = -1 \cdot \sin(2 \pi \cdot 5 \cdot t + \frac{\pi}{2})$')

# Subplot 4: y4
plt.subplot(2, 2, 4)
plt.plot(t, y4)
plt.title(r'$y_4 = 0.5 \cdot \cos(2 \pi \cdot 6 \cdot t + \pi)$')

# Mengatur layout agar tidak tumpang tindih
plt.tight_layout()
plt.show()
```





3. Pertanyaan Analisis:

In [101]...

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

# Membuat array t dari 0 hingga 2 dengan langkah 0.0001
t = np.arange(0, 2, 0.0001)

# Membuat sinyal-sinyal
y1 = 2 * np.sin(2 * np.pi * 3 * t + 0)
y2 = 1 * np.cos(2 * np.pi * 4 * t + np.pi / 4)
y3 = -1 * np.sin(2 * np.pi * 5 * t + np.pi / 2)
y4 = 0.5 * np.cos(2 * np.pi * 6 * t + np.pi)

# Membuat plot untuk setiap sinyal
plt.figure(figsize=(10, 8))

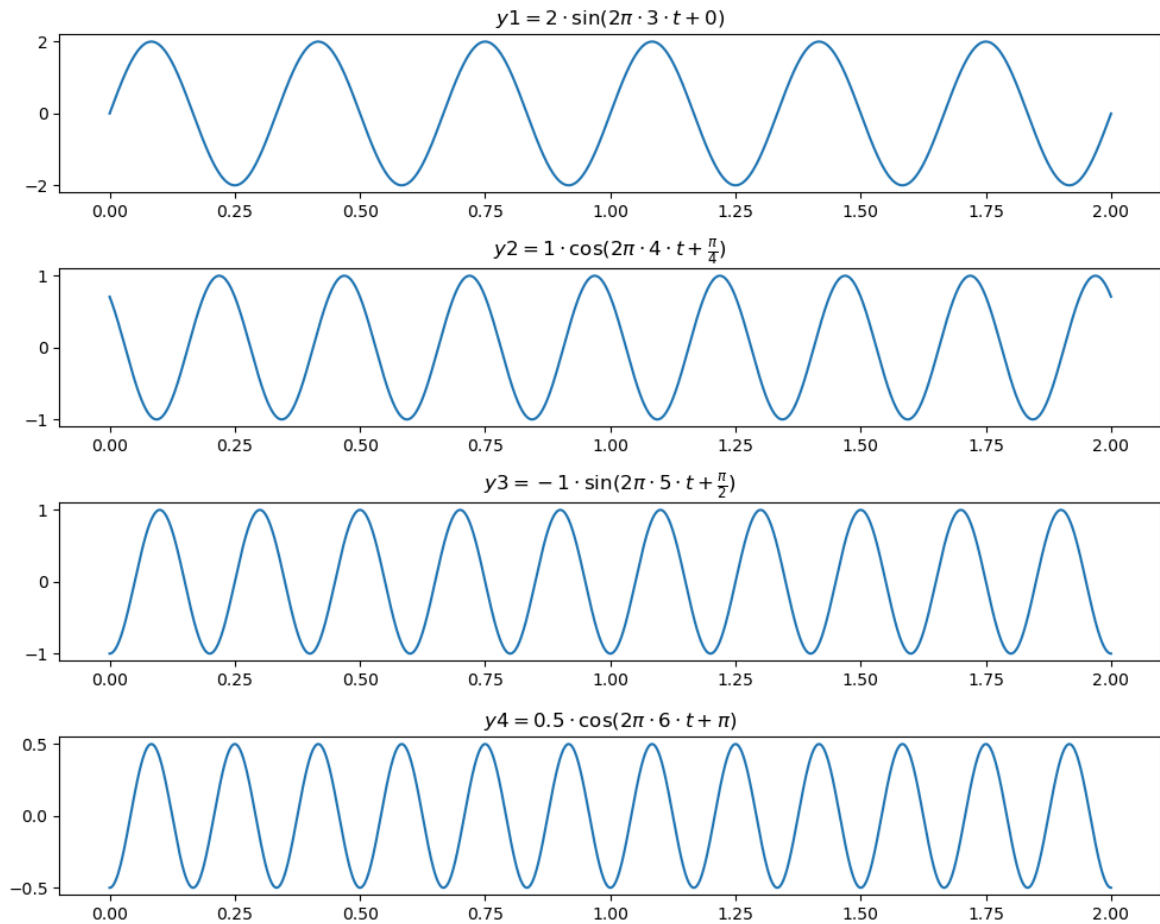
# Plot y1
plt.subplot(4, 1, 1)
plt.plot(t, y1)
plt.title(r'$y1 = 2 \cdot \sin(2 \pi \cdot 3 \cdot t + 0)$')

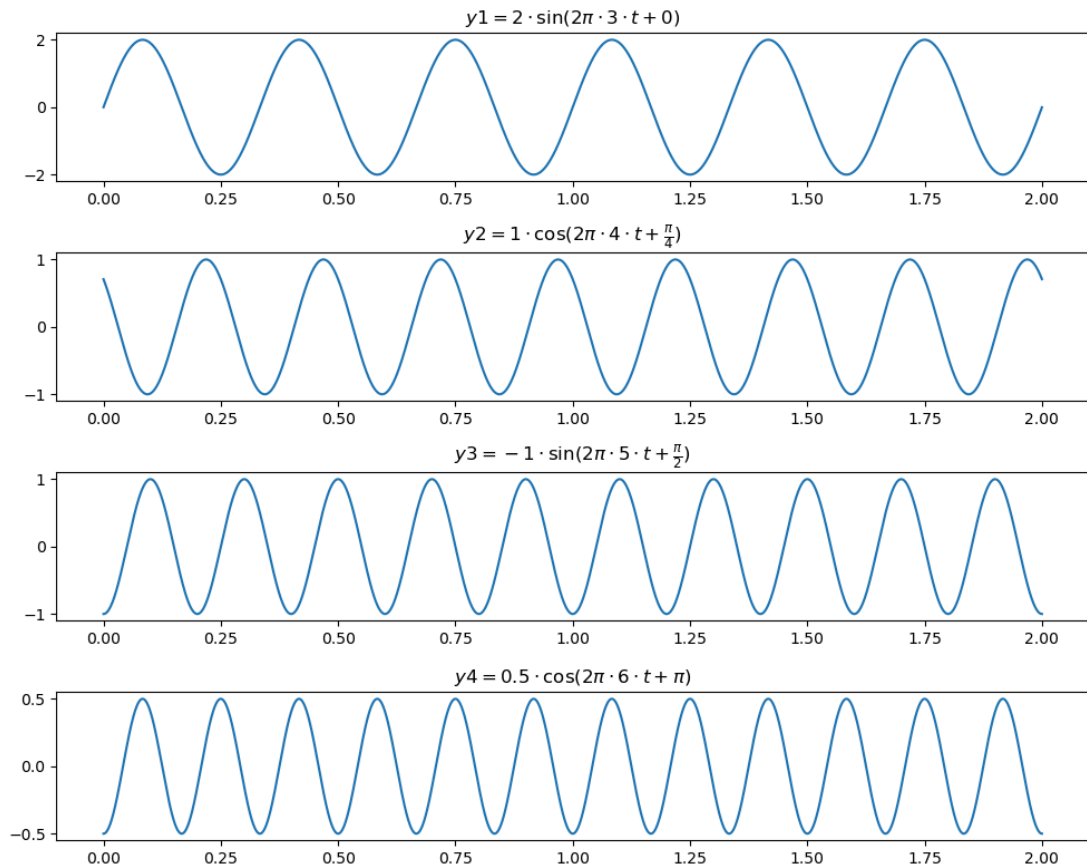
# Plot y2
plt.subplot(4, 1, 2)
plt.plot(t, y2)
plt.title(r'$y2 = 1 \cdot \cos(2 \pi \cdot 4 \cdot t + \frac{\pi}{4})$')

# Plot y3
plt.subplot(4, 1, 3)
plt.plot(t, y3)
plt.title(r'$y3 = -1 \cdot \sin(2 \pi \cdot 5 \cdot t + \frac{\pi}{2})$')
```

```
# Plot y4
plt.subplot(4, 1, 4)
plt.plot(t, y4)
plt.title(r'$y_4 = 0.5 \cdot \cos(2 \pi \cdot 6 \cdot t + \pi)$')

# Mengatur layout agar tidak tumpang tindih
plt.tight_layout()
plt.show()
```





A. Berapa amplitudo dan frekuensi masing-masing sinyal?:

- **Sinyal y1:**
 - Amplitudo: 2 (dari persamaan ($y1 = 2 \cdot \sin(2\pi \cdot 3 \cdot t + 0)$))
 - Frekuensi: 3 Hz (karena frekuensi di dalam persamaan adalah 3)
- **Sinyal y2:**
 - Amplitudo: 1 (dari persamaan ($y2 = 1 \cdot \cos(2\pi \cdot 4 \cdot t + \frac{\pi}{4})$))
 - Frekuensi: 4 Hz (karena frekuensi di dalam persamaan adalah 4)
- **Sinyal y3:**
 - Amplitudo: 1 (dari persamaan ($y3 = -1 \cdot \sin(2\pi \cdot 5 \cdot t + \frac{\pi}{2})$))
 - Frekuensi: 5 Hz (karena frekuensi di dalam persamaan adalah 5)
- **Sinyal y4:**
 - Amplitudo: 0.5 (dari persamaan ($y4 = 0.5 \cdot \cos(2\pi \cdot 6 \cdot t + \pi)$))
 - Frekuensi: 6 Hz (karena frekuensi di dalam persamaan adalah 6)

B. Bagaimana pergeseran fase mempengaruhi posisi gelombang?:

- Pergeseran fase menentukan posisi awal dari gelombang. Misalnya, dalam sinyal ($y2$) dan ($y4$), kita memiliki pergeseran fase sebesar ($\frac{\pi}{4}$) dan (π) berturut-turut.
- Pergeseran fase positif menggeser gelombang ke kiri, dan nilai negatif menggeser gelombang ke kanan.

- Dalam sinyal (y_2), fase sebesar ($\frac{\pi}{4}$) menyebabkan gelombang dimulai sedikit lebih awal.
- Dalam sinyal (y_4), fase sebesar (π) membuat gelombang mulai dari posisi yang berlawanan.

Anda dapat bereksperimen dengan mengubah nilai fase untuk melihat bagaimana ini menggeser posisi gelombang.

C. Bandingkan sinyal-sinyal dengan amplitudo yang berbeda dan diskusikan bagaimana amplitudo mempengaruhi tampilan gelombang.:

- Amplitudo menentukan **tinggi** atau **intensitas** dari gelombang. Semakin besar amplitudo, semakin tinggi puncak dan lembah gelombang.
- Misalnya, sinyal (y_1) memiliki amplitudo terbesar (2), sehingga puncak dan lembah gelombang lebih tinggi dibandingkan dengan sinyal lainnya.
- Sinyal (y_4) memiliki amplitudo terkecil (0.5), sehingga puncak dan lembah gelombang jauh lebih pendek dibandingkan sinyal lainnya.
- Amplitudo hanya mempengaruhi tinggi gelombang, tetapi tidak mengubah frekuensi atau posisi.

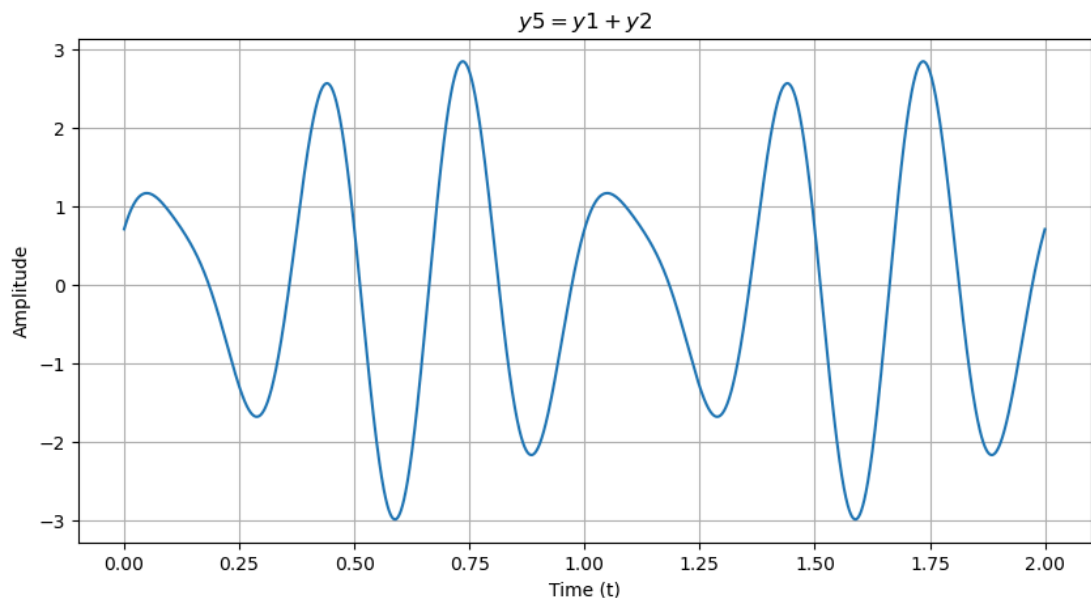
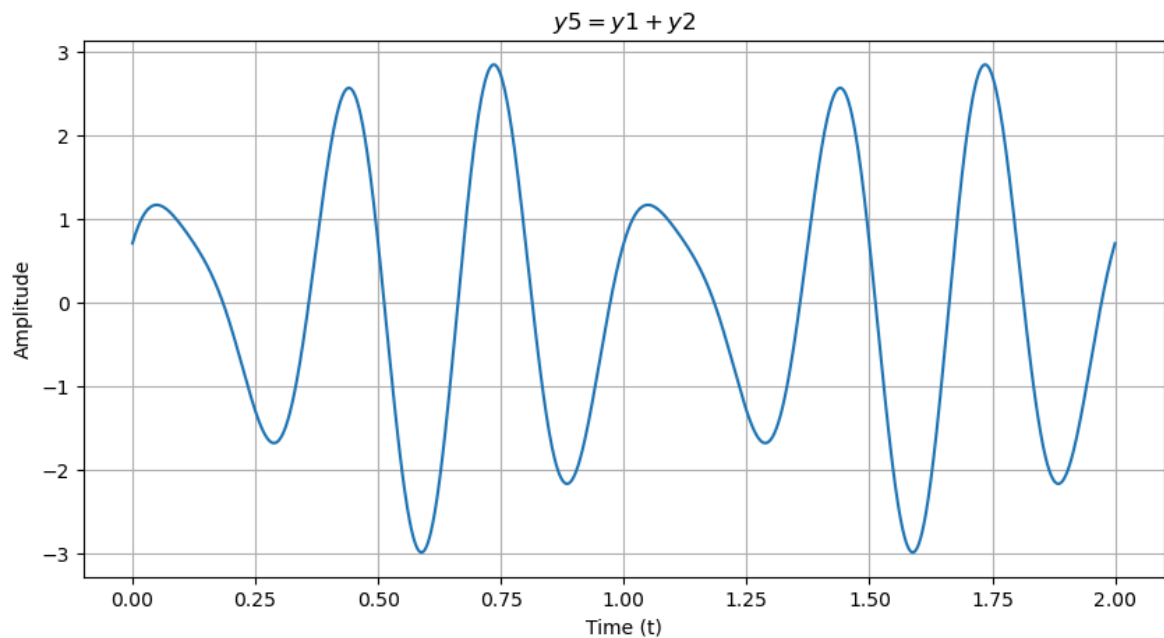
D. Bandingkan sinyal-sinyal dengan pergeseran fase yang berbeda dan diskusikan bagaimana pergeseran fase mempengaruhi tampilan gelombang.:

- Pergeseran fase menentukan **di mana** gelombang dimulai. Misalnya, pada sinyal (y_2) dan (y_4), pergeseran fase ($\frac{\pi}{4}$) dan (π) menggeser posisi gelombang.
- Sinyal dengan fase lebih besar terlihat seperti bergeser ke kiri, sedangkan yang tanpa fase tambahan dimulai di titik nol.
- Dengan kata lain, pergeseran fase mengatur waktu awal osilasi gelombang, namun tidak mengubah tinggi (amplitudo) atau seberapa sering gelombang berosilasi (frekuensi).

4. Tugas Lanjutan:

```
In [102... # Membuat sinyal baru y5 yang merupakan kombinasi dari y1 dan y2
y5 = y1 + y2

# Plot y5
plt.figure(figsize=(10, 5))
plt.plot(t, y5)
plt.title(r'$y_5 = y_1 + y_2$')
plt.xlabel('Time (t)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.show()
```

Diskusi Kombinasi Dua Gelombang

1. Superposisi Gelombang:

- Ketika dua gelombang digabungkan (dalam hal ini y_1 dan y_2), hasilnya adalah superposisi atau penjumlahan dari kedua gelombang tersebut.
- Karena y_1 adalah gelombang sinus dengan frekuensi 3 Hz dan y_2 adalah gelombang cosinus dengan frekuensi 4 Hz, bentuk gelombang y_5 merupakan kombinasi dari karakteristik kedua gelombang tersebut.

2. Pengaruh Amplitudo dan Frekuensi:

- Frekuensi kedua gelombang berbeda (3 Hz dan 4 Hz), sehingga kombinasi mereka menghasilkan gelombang dengan pola yang kompleks (terlihat berosilasi

dengan cepat).

- Amplitudo dari y_1 (2) lebih besar daripada y_2 (1), sehingga kontribusi dari y_1 lebih dominan dalam membentuk puncak dan lembah dari y_5 .

3. Pergeseran Fase:

- Pergeseran fase pada y_2 menyebabkan kedua gelombang tidak sejajar sempurna pada titik awal. Ini menciptakan sedikit pergeseran dalam hasil akhir y_5 , yang terlihat dari bagaimana puncak dan lembah gelombang saling bertumpuk atau membatalkan sebagian satu sama lain.

Bantuan jawaban dari pertanyaan yang ada di Hads On 1 Tugas 1-4 ini dari Chat Gpt, dan inilah link yang saya tanyakan dan saya pelajari dari :
<https://chatgpt.com/share/66f7ec21-511c-8007-81b6-ceb78cd1435d>

Hands On 1

Melanjutkan Instruksi Tugas

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

Tugas 5: Downsampling Sinyal ECG dan Aliasing:

Install Library Yang Belum Ada

```
!pip install neurokit2
```

Import Library

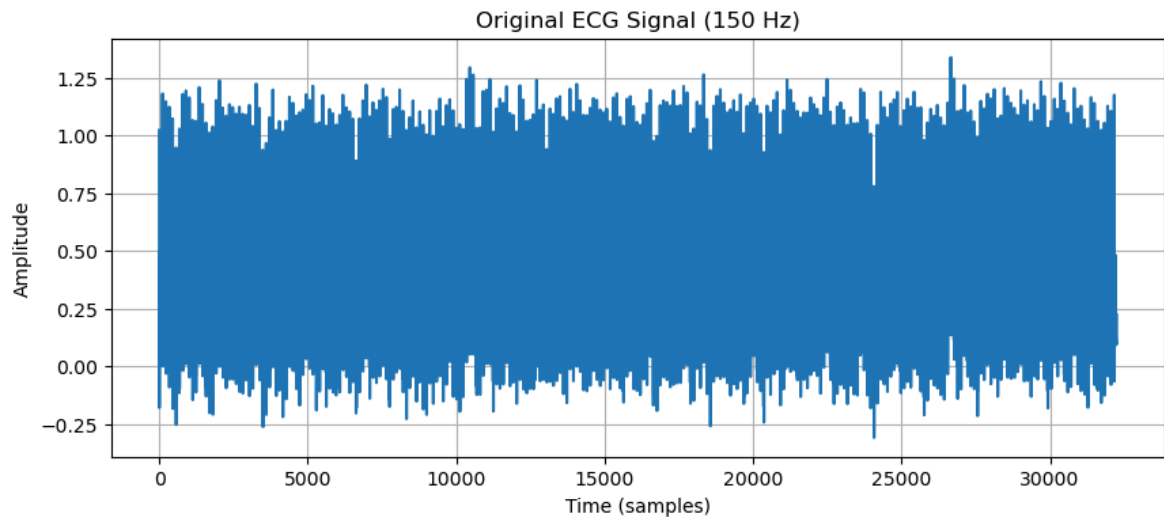
```
In [103... import numpy as np
import neurokit2 as nk
import matplotlib.pyplot as plt
from scipy.signal import butter, filtfilt, resample
```

1. Simulasi Sinyal ECG

```
In [104... # Spesifikasi Sinyal ECG
nim = '121140215' # NIM Saya
durasi = int(nim[-3:]) # Berdasarkan 3 digit terakhir NIM Saya
sampling_rate = 150 # Hz
noise_level = float(f"0.{nim[-2:]}") # Berdasarkan 2 digit terakhir NIM
heart_rate = 80 # BPM
random_state = 20030520 # Tanggal lahir Saya 20 Mei 2003 dengan format Y

# Generate ECG Signal
ecg_signal = nk.ecg_simulate(duration=durasi, sampling_rate=sampling_rate,
                             heart_rate=heart_rate, noise=noise_level,
                             random_state=random_state)

# Plot Original Signal
plt.figure(figsize=(10, 4))
plt.plot(ecg_signal)
plt.title("Original ECG Signal (150 Hz)")
plt.xlabel("Time (samples)")
plt.ylabel("Amplitude")
plt.grid(True)
plt.show()
```

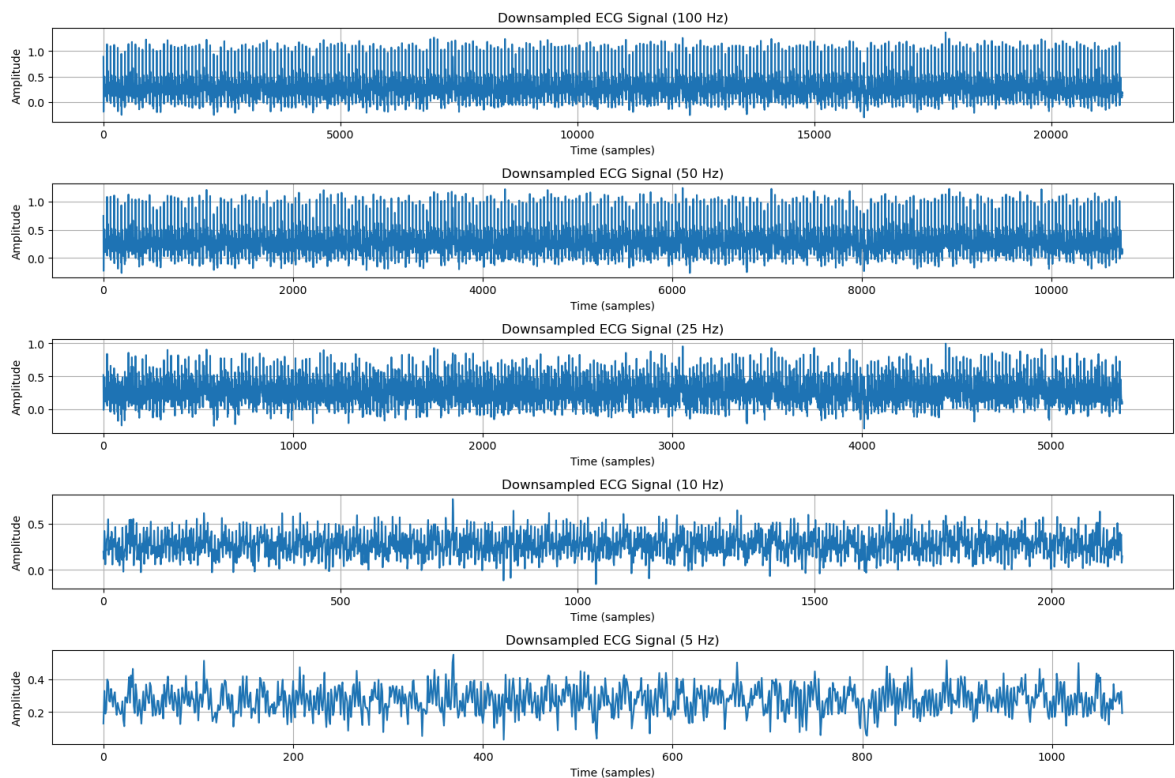


2. Downsampling ke Beberapa Frekuensi Sampling

```
In [105.. # Downsampling Rates
downsampling_rates = [100, 50, 25, 10, 5]

# Function for Downsampling
def downsample_signal(signal, original_rate, new_rate):
    num_samples = int(len(signal) * new_rate / original_rate)
    return resample(signal, num_samples)

# Downsample and Plot
plt.figure(figsize=(15, 10))
for i, rate in enumerate(downsampling_rates):
    downsampled_signal = downsample_signal(ecg_signal, sampling_rate, rate)
    plt.subplot(len(downsampling_rates), 1, i + 1)
    plt.plot(downsampled_signal)
    plt.title(f"Downsampled ECG Signal ({rate} Hz)")
    plt.xlabel("Time (samples)")
    plt.ylabel("Amplitude")
    plt.grid(True)
plt.tight_layout()
plt.show()
```



3. Penjelasan Aliansi

- Aliasing adalah fenomena dalam pengolahan sinyal digital di mana sinyal yang direkam atau diproses tidak dapat menangkap perubahan frekuensi secara akurat, sehingga muncul distorsi atau kesalahan dalam representasi sinyal tersebut. Ini terjadi ketika sinyal analog dikonversi menjadi sinyal digital tanpa memenuhi syarat teorema Nyquist, yaitu frekuensi sampling harus minimal dua kali frekuensi tertinggi dari sinyal asli.
- Penjelasan sederhana: Bayangkan kamu merekam suara yang memiliki nada tinggi dengan mikrofon yang mengambil sampel terlalu sedikit per detik. Akibatnya, mikrofon tidak dapat menangkap perubahan nada tersebut dengan benar, dan ketika suara itu diputar ulang, nada yang direkam terdengar berbeda atau seperti suara 'bergoyang.' Itulah yang disebut dengan aliasing.
- Contoh umum: Aliasing sering terjadi dalam fotografi digital, seperti saat memotret pola garis yang rapat (misalnya, kisi-kisi atau pakaian bergaris tipis). Kamera tidak dapat menangkap pola tersebut dengan benar, sehingga muncul efek "moire," di mana pola garis terlihat berubah atau bergelombang.

Tugas 6: Eksperimen dengan Parameter order pada Filter

1. Penjelasan Order pada Filter

- Order dari filter menunjukkan jumlah poles atau derajat filter tersebut. Semakin tinggi order, semakin tajam transisi antara passband dan stopband, namun juga meningkatkan kompleksitas komputasi dan dapat menyebabkan fase sinyal berubah lebih drastis.

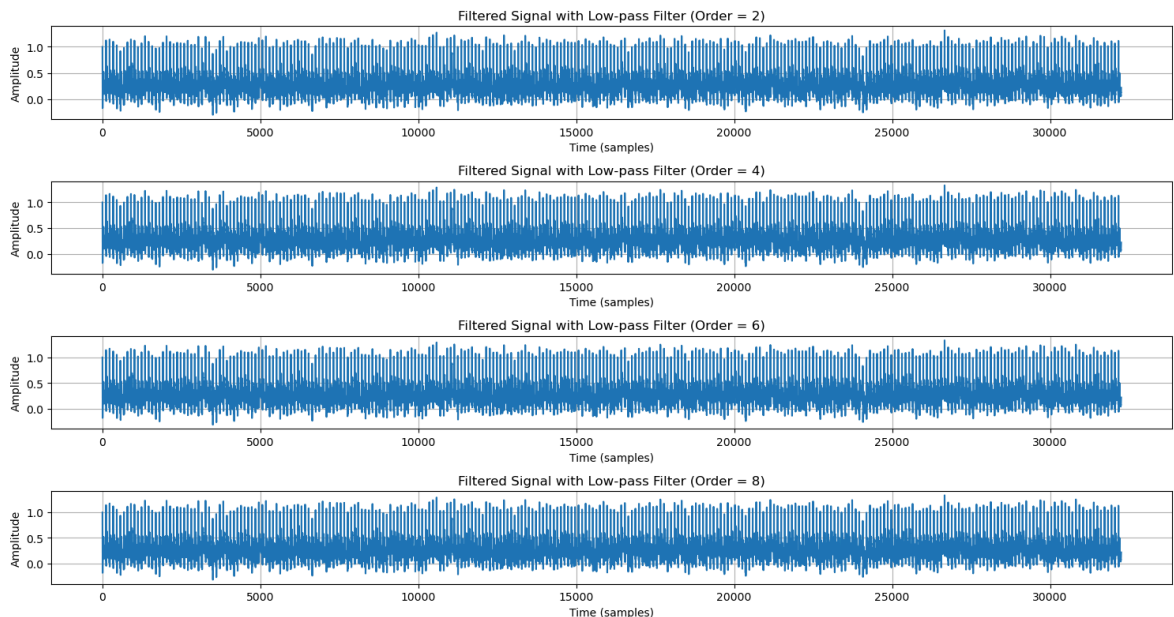
2. Eksperimen dengan Filter Low-pass

```
In [106... # Sinyal dengan Noise
noisy_signal = ecg_signal + 0.05 * np.random.randn(len(ecg_signal))

# Filter Function
def lowpass_filter(signal, cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = butter(order, normal_cutoff, btype='low', analog=False)
    return filtfilt(b, a, signal)

# Filter with Different Orders
orders = [2, 4, 6, 8]

plt.figure(figsize=(15, 8))
for i, order in enumerate(orders):
    filtered_signal = lowpass_filter(noisy_signal, 30, sampling_rate, order)
    plt.subplot(len(orders), 1, i + 1)
    plt.plot(filtered_signal)
    plt.title(f"Filtered Signal with Low-pass Filter (Order = {order})")
    plt.xlabel("Time (samples)")
    plt.ylabel("Amplitude")
    plt.grid(True)
plt.tight_layout()
plt.show()
```



3. Penjelasan Pengaruh Order

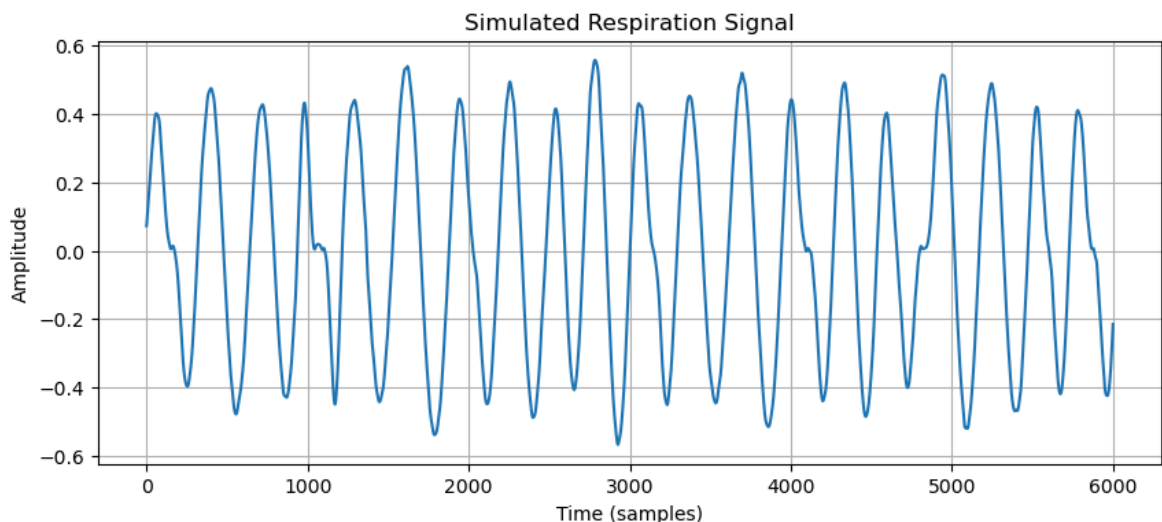
- Dengan meningkatkan order, filter menjadi lebih tajam dalam mengurangi komponen frekuensi tinggi, tetapi dapat memperbesar efek fase non-linier dan membutuhkan lebih banyak komputasi.

Tugas 7: Filter Band-pass pada Sinyal Respirasi

1. Simulasi Sinyal Respirasi

```
In [107... # Simulasi Sinyal Respirasi menggunakan nk.rsp_simulate
respiration_signal = nk.rsp_simulate(duration=60, sampling_rate=100, resp

# Plot Sinyal Respirasi
plt.figure(figsize=(10, 4))
plt.plot(respiration_signal)
plt.title("Simulated Respiration Signal")
plt.xlabel("Time (samples)")
plt.ylabel("Amplitude")
plt.grid(True)
plt.show()
```



- disini saya mencoba "pip install neurokit2 --upgrade" karna terjadi nya error :

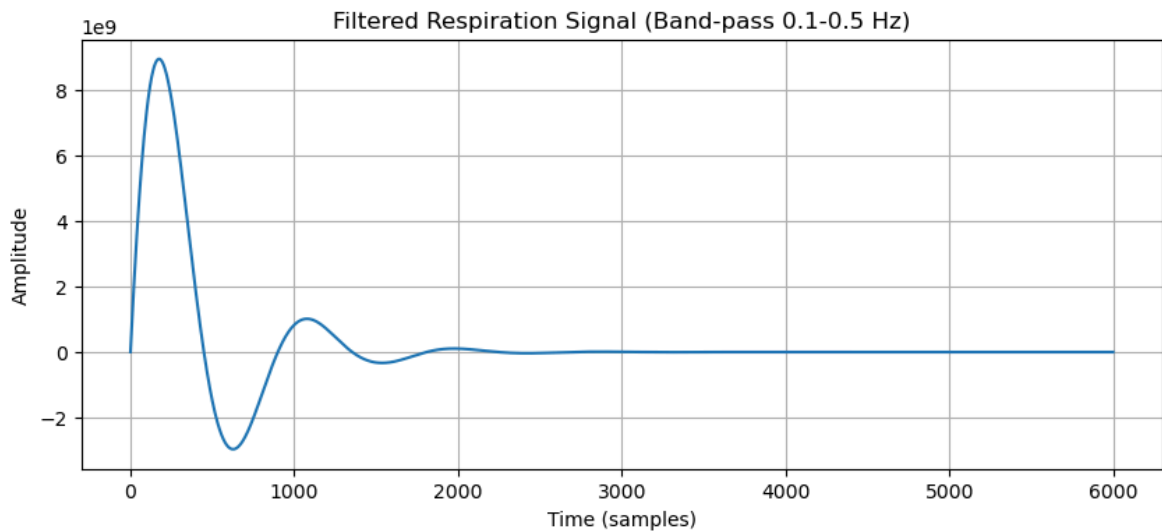
2. Desain Filter Band-pass

```
In [108... # Filter Function
def bandpass_filter(signal, lowcut, highcut, fs, order=4):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return filtfilt(b, a, signal)

# Apply Band-pass Filter
lowcut = 0.1 # Minimum cutoff frequency in Hz
highcut = 0.5 # Maximum cutoff frequency in Hz
filtered_respiration = bandpass_filter(respiration_signal, lowcut, highcu

# Plot Filtered Signal
plt.figure(figsize=(10, 4))
```

```
plt.plot(filtered_respiration)
plt.title(f"Filtered Respiration Signal (Band-pass {lowcut}-{highcut} Hz)")
plt.xlabel("Time (samples)")
plt.ylabel("Amplitude")
plt.grid(True)
plt.show()
```



3. Penjelasan Pemilihan Frekuensi Cutoff

- Frekuensi pernapasan manusia biasanya berada di rentang 0.1 - 0.5 Hz (6 - 30 BPM). Oleh karena itu, cutoff pada 0.1 Hz dan 0.5 Hz dipilih untuk mengisolasi komponen frekuensi yang relevan dari sinyal respirasi dan menghilangkan noise atau gangguan di luar rentang tersebut.

Bantuan jawaban dari pertanyaan yang ada di Hads On 1 Tugas 5-7 ini dari Chat Gpt, dan inilah link yang saya tanyakan dan saya pelajari dari :

<https://chatgpt.com/share/67054cab-51e8-8007-bb04-8e35ddeaa64b>