Preprocessing Sinyal Suara Jantung (PCG)

by Jusak Jusak

Submission date: 05-May-2021 01:44PM (UTC+0700)
Submission ID: 1578488664
File name: Lampiran_HKI_Preprocessing_rev.pdf (664.36K)
Word count: 793
Character count: 4340

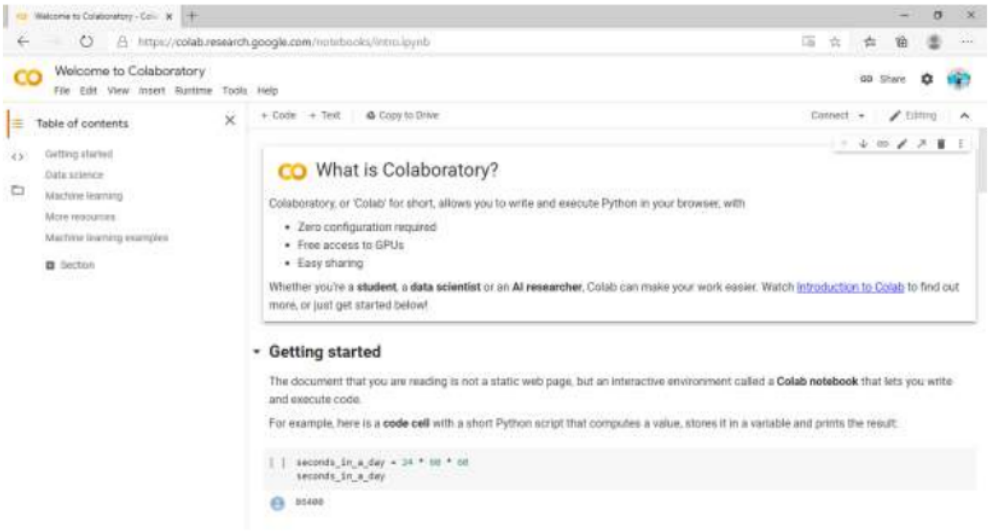
Pre-Processing Sinyal Suara Jantung (PCG)

Dibuat oleh :
1. Jusak
2. Ira Puspasari
3. Weny Indah Kusumawati
4. Rizky Santya Pradana
5. Muhammad Gerald Rizky
6. Miskiyanto
7. Gusti Rafi Afkariansyah

MANUAL BOOK MENJALANKAN PROGRAM:

Berikut ini adalah cara menjalankan Program Pre-Processing dengan menggunakan bahasa pemrograman Python. Langkah-langkah yang harus dilakukan adalah:

1. Buka colab.research.google.com, seperti ditunjukkan pada Gambar 1.



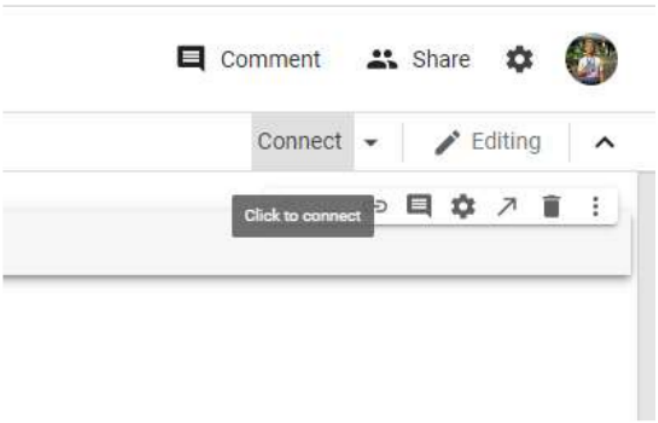
Gambar 1. Membuka Google Collabs

2. Klik File, New notebook, akan terlihat seperti Gambar 2.

Gambar 2. Memulai Project baru

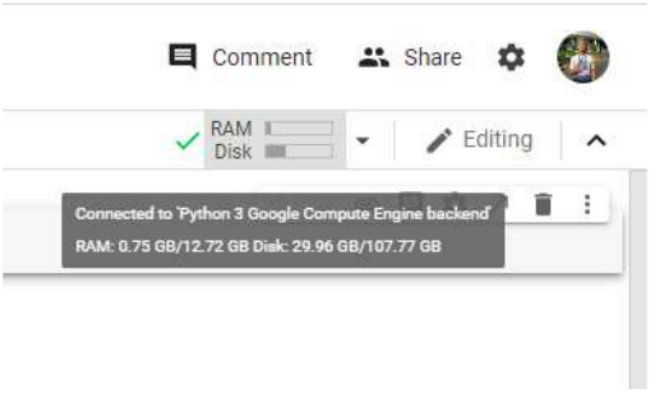
3. Hasil dari Gambar 2, akan terlihat seperti Gambar 3. Akan muncul jendela baru.

4. Langkah berikutnya melakukan koneksi ke Cloud, tampak seperti Gambar 4.



Gambar 4. Klik Connect untuk proses penyambungan

5. Langkah berikutnya setelah terkoneksi, tampak seperti Gambar 5.



Gambar 5. Keadaan setelah terkoneksi

6. Langkah berikutnya adalah menginstal beberapa library:

Import library yg digunakan

Numpy: Untuk Operasi Vektor dan Matriks

Matplotlib: Visualisasi plot grafik

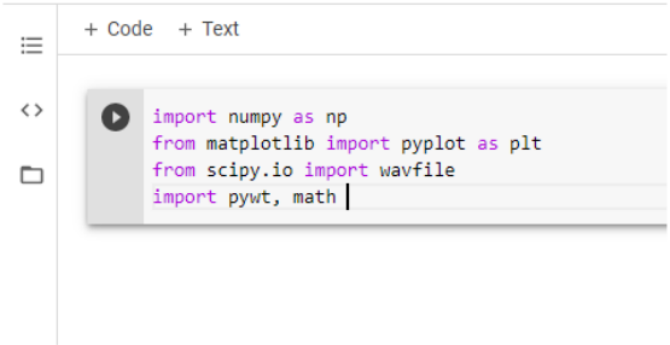
Scipy: Operasi aljabar dan matrix

Pywt: Untuk wavelet transform

Math: Untuk fungsi matematika

Run cell (Ctrl + Enter) untuk menjalankan perintah

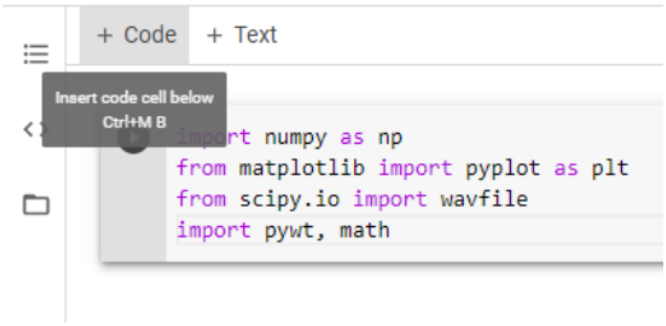
Proses penambahan library dapat dilihat pada Gambar 6.



Gambar 6. Proses Import library

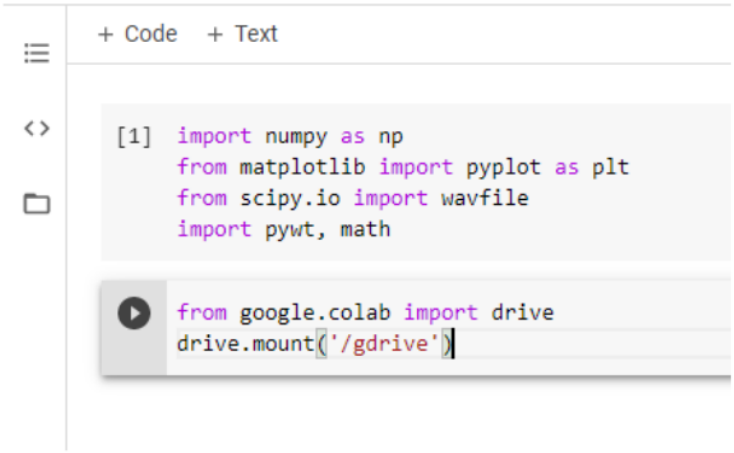
7. Tambahkan code cell (Ctrl + M + B) untuk menulis syntax perintah lainnya

Memisah code cell bertujuan untuk mempermudah melacak syntax error. Seperti pada Gambar 7.



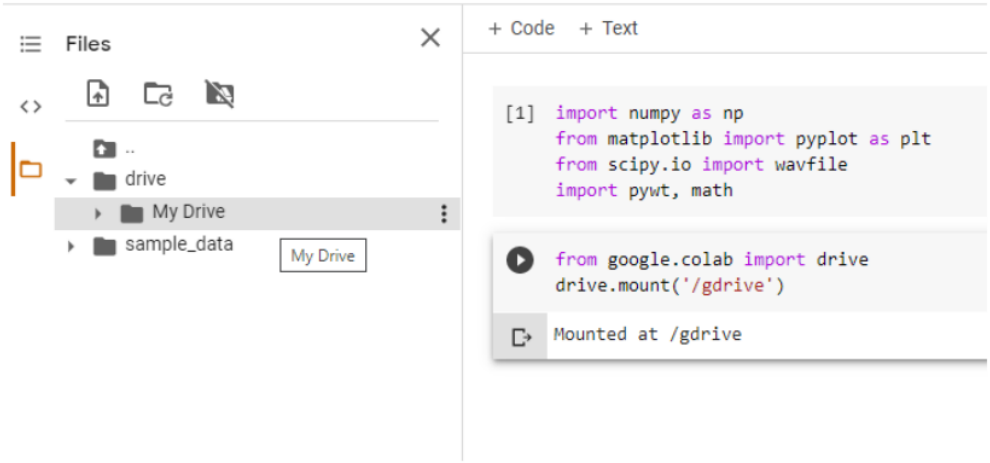
Gambar 7. Memilih syntax lainnya

8. Menghubungkan Google Colab dengan Google Drive untuk mengakses data, seperti pada Gambar 8.



Gambar 8. Terhubung dengan Google Drive

9. Setelah terhubung, tampak pada Gambar 9.



Gambar 9. Tampak collabs terhubung dengan drive

10. Jika sudah terhubung ke Google Drive, Untuk mengakses file b0001.wav di google drive pada direktori /My Drive/Colab Notebooks/sinyal_b/ ditunjukkan pada Gambar 10.

Gambar 10. Terhubung dengan Drive

11. Untuk mengetahui total waktu file wav yg sudah diakses, seperti pada Gambar 11.

```
t_start = 0
t_total = len(data)/fs
print("total time : ", t_total, "s")
t=[]
n = 0
while (n < len(data)):
    if n==0 :
        t.append(t_start + 1/fs)
    else :
        t.append(t[n-1] + 1/fs)
    n = n + 1
```

```
total time :  8.0 s
```

Gambar 11. Total waktu File Wav

12. Tahapan mengetahui nilai fs dan jumlah data amplitudo, ditunjukkan pada Gambar 12.

```
maxData = np.amax(data)
dataPCG = data/maxData
L = dataPCG.shape[0]
print("Type of dataPCG",type(dataPCG))
print("fs is ", fs)
print("L is ", L)
```

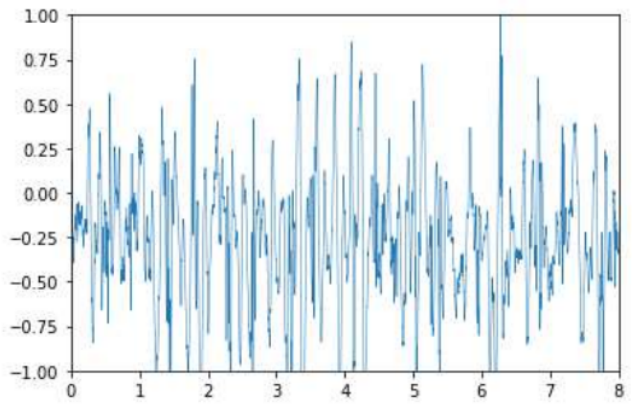
```
Type of dataPCG <class 'numpy.ndarray'>
fs is  2000
L is  16000
The shape of dataPCG is  (16000,)
```

Gambar 12. Nilai fs dan jumlah data amplitude

13. Untuk menggambarkan sinyal PCG, ditunjukkan pada Gambar 13.

```
plt.figure(1)
plt.plot(t,dataPCG,linewidth=0.5)
plt.axis([0, 8, -1, 1])
```

```
(0.0, 8.0, -1.0, 1.0)
```



Gambar 13. Plotting sinyal PCG

14. Memberi batasan sumbu X dan Y , seperti pada Gambar 14.

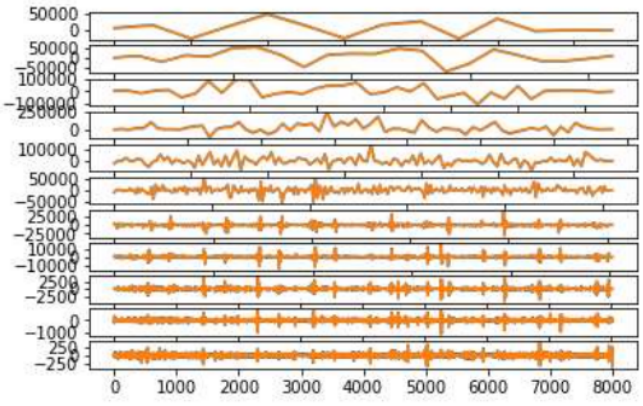
```
w = pywt.Wavelet('sym4')
maxlev = pywt.dwt_max_level(len(data), w.dec_len)
print("maximum level is " + str(maxlev))
threshold = 0.04

coeffs = pywt.wavedec(data, 'sym4', level=maxlev)
```

```
maximum level is 11
```

Gambar 14. Proses cek limit sumbu X dan Y

15. Plotting wavelet untuk proses thresholding, pada Gambar 15.

```
plt.figure()
for i in range(1, len(coeffs)):
    plt.subplot(maxlev, 1, i)
    plt.plot(coeffs[i])
    coeffs[i] = pywt.threshold(coeffs[i], threshold*max(coeffs[i]))
    plt.plot(coeffs[i])
```



Gambar 15. Plot setiap orde wavelet

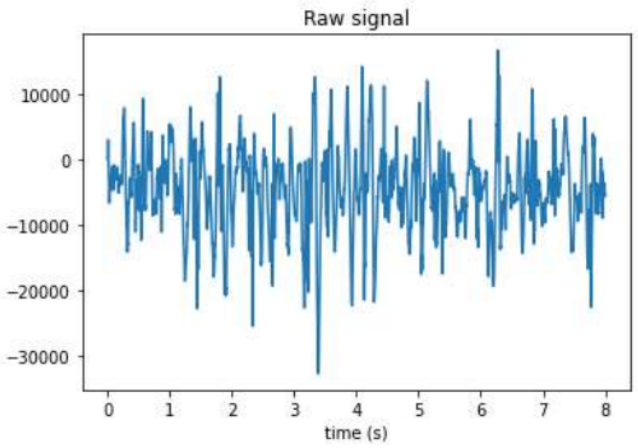
16. Proses wavelet dengan mother wavelet symlet orde 4, dengan batasan data 0-16.000 data amplitude. Seperti pada Gambar 16.

```
datarec = pywt.waverec(coeffs, 'sym4')
mintime = 0
maxtime = mintime + 15999
```

Gambar 16. Proses wavelet dengan 16.000 data

17. Proses memploting sinyal PCG asli dan sinyal PCG yg sudah di proses Denoise. Seperti ditunjukkan pada Gambar 17.

```
plt.figure()
plt.plot()
plt.plot(t[mintime:maxtime], data[mintime:maxtime])
plt.xlabel('time (s)')
plt.title("Raw signal")
plt.figure()
plt.plot()
plt.plot(t[mintime:maxtime], datarec[mintime:maxtime])
plt.xlabel('time (s)')
plt.title("De-noised signal using wavelet techniques")
plt.show()
```



Gambar 17. Plotting gambar denoising

18. Untuk menyimpan nilai waktu dan nilai sinyal yg sudah di denoise, gunakan langkah seperti pada Gambar 18.

```
TData = t[mintime:maxtime]
SDen = datarec[mintime:maxtime]
```

Gambar 17. Nilai Waktu

LAMPIRAN PROGRAM:

```
#!/usr/bin/env python
# coding: utf-8


# In[30]:




import numpy as np

from matplotlib import pyplot as plt

from scipy.io import wavfile

import pywt, math


# Commented out IPython magic to ensure Python compatibility.

from google.colab import drive

drive.mount('/gdrive')


# load data

(fs, data) = wavfile.read('/gdrive/My Drive/Colab Notebooks/sinyal_b/b0001.wav') # using Scipy


t_start = 0

t_total = len(data)/fs

print("total time : ", t_total, "s")

t=[]

n = 0

while (n < len(data)):

   if n==0 :

      t.append(t_start + 1/fs)
```

```
    else :
        t.append(t[n-1] + 1/fs)
    #print("t[", n, "] : ", round(t[0,n],5))
    n = n + 1

# Data normalization
maxData = np.amax(data)
dataPCG = data/maxData
L = dataPCG.shape[0]
#print("Type of dataPCG",type(dataPCG))
#print("fs is ", fs)
#print("L is ", L)
#print("The shape of dataPCG is ", dataPCG.shape)

# time
#t = np.array([np.linspace(start=0, stop=L/fs, num=L, dtype=float)])
#print("The shape of t is ", t.shape)

# Plot PCG signal
plt.figure(1)
plt.plot(t,dataPCG,linewidth=0.5)
plt.axis([0, 8, -1, 1])



# In[34]:
```

```
# Create wavelet object and define parameters
w = pywt.Wavelet('sym4')
maxlev = pywt.dwt_max_level(len(data), w.dec_len)
# maxlev = 2 # Override if desired
print("maximum level is " + str(maxlev))
threshold = 0.04 # Threshold for filtering

# Decompose into wavelet components, to the level selected:
coeffs = pywt.wavedec(data, 'sym4', level=maxlev)

#cA = pywt.threshold(cA, threshold*max(cA))
plt.figure()
for i in range(1, len(coeffs)):
    plt.subplot(maxlev, 1, i)
    plt.plot(coeffs[i])
    coeffs[i] = pywt.threshold(coeffs[i], threshold*max(coeffs[i]))
    plt.plot(coeffs[i])

datarec = pywt.waverec(coeffs, 'sym4')

mintime = 0
maxtime = mintime + 15999

plt.figure()
plt.plot()
```

```
plt.plot(t[mintime:maxtime], data[mintime:maxtime])
plt.xlabel('time (s)')
plt.ylabel('microvolts (uV)')
plt.title("Raw signal")
plt.figure()
plt.plot()
plt.plot(t[mintime:maxtime], datarec[mintime:maxtime])
plt.xlabel('time (s)')
plt.ylabel('microvolts (uV)')
plt.title("De-noised signal using wavelet techniques")


#plt.tight_layout()
plt.show()


TData = t[mintime:maxtime]
SDen = datarec[mintime:maxtime]



# In[ ]:
```

Preprocessing Sinyal Suara Jantung (PCG)

ORIGINALITY REPORT

12%	7%	3%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Nazarbayev University Student Paper	3%
2	Submitted to University of Bucharest Student Paper	2%
3	hanyeah.com Internet Source	2%
4	coderlessons.com Internet Source	2%
5	es.scribd.com Internet Source	1%
6	www.scribd.com Internet Source	1%
7	Andreas François Vermeulen. "Practical Data Science", Springer Science and Business Media LLC, 2018 Publication	1%

Exclude quotes	On	Exclude matches	Off

Exclude bibliography Off