



Nama: **Aprian Yusuf Nugroho (118140144)**

Tugas Ke: **3**

Mata Kuliah: **Pengolahan Sinyal Digital (IF3024)**

Tanggal: **23 Desember 2024**

Final Project Digital Signal Processing (IF3024)

Laporan ini berisi hasil percobaan pembuatan sistem pengukuran sinyal respirasi dan sistem pengukuran remote-photoplethysmography (rPPG). Sistem ini menerima input berupa video dari webcam, yang kemudian secara real-time memproses video dan menampilkan sinyal respirasi dan sinyal rPPG.

1 Persiapan Pengerjaan

1.1 Install Python

Install Python versi 3.8 atau lebih baru

1.2 Setup Virtual Environment

- Buat folder, lalu buka terminal dan jalankan
- `python -m venv venv` (Buat virtual environment)
- `.\venv\Scripts\activate` (Aktifkan virtual environment)
- `pip install --upgrade pip setuptools` (Update pip dan setuptools)

1.3 Install Dependencies

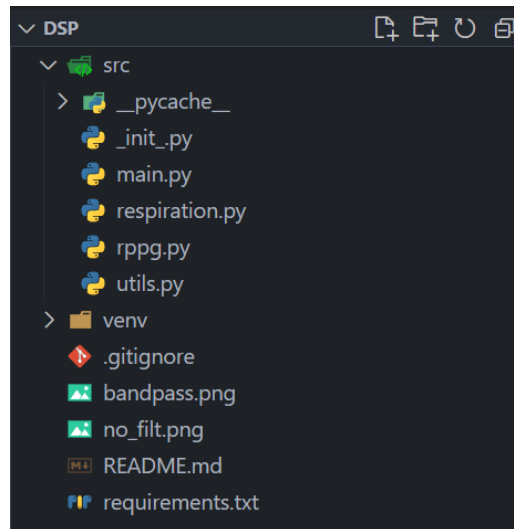
Install library yang dibutuhkan *"pip install opencv-python numpy scipy matplotlib"*

1.4 Install Dependencies

Buat File Requirements, kemudian simpan dependencies kedalam file requirement.txt *"pip freeze > requirements.txt"*

2 Implementasi Kode

Sebelum melakukan pengkodean, buat struktur programnya terlebih dahulu



Gambar 1: Struktur Program

2.1 main.py

File ini adalah program utama untuk memproses video dari webcam dan menampilkan sinyal respirasi serta rPPG. Berikut isi dari file *main.py*

```
1  import cv2
2  from respiration import ekstrak_sinyal_respirasi
3  from rppg import ekstrak_sinyal_rppg
4  from utils import butterworth_filter, bandpass_filter
5  import matplotlib.pyplot as plt
6  import sys
7  import os
8
9  # Menambahkan path direktori 'src' ke sys.path
10 sys.path.append(os.path.abspath(os.path.join(
11     os.path.dirname(__file__), "..", "src")))
12
13 def main():
14     # Membuka webcam
15     cap = cv2.VideoCapture(0)
16     if not cap.isOpened():
17         print("Error: Tidak dapat mengakses webcam.")
18         return
19
20     # Membuat plot untuk visualisasi sinyal
21     plt.ion()
22     fig, (ax1, ax2) = plt.subplots(2, 1)
23     sinyal_respirasi, sinyal_rppg = [], []
24     fig.tight_layout(pad=4)
25
26     # Buat kondisi perulangan untuk menjalankan program
27     while True:
28         # Membaca frame dari webcam
29         ret, frame = cap.read()
30         if not ret:
31             break
32
33         # Mengubah ukuran frame untuk mengurangi beban komputasi
34         frame_resized = cv2.resize(frame, (320, 240))
35
36         # Ekstraksi sinyal respirasi dan rPPG
```

```

37     resp_signal = ekstrak_sinyal_respirasi(frame_resized)
38     rppg_signal = ekstrak_sinyal_rppg(frame_resized)
39
40     # Simpan sinyal
41     sinyal_respirasi.append(resp_signal)
42     sinyal_rppg.append(rppg_signal)
43
44     # Menampilkan video dari webcam
45     cv2.imshow("Webcam", frame_resized)
46
47     # Memperbarui plot sinyal
48     ax1.clear()
49     ax1.plot(sinyal_respirasi[-100:], color='blue')
50     ax1.set_title("Sinyal Respirasi")
51     ax1.set_xlabel("Waktu")
52     ax1.set_ylabel("Amplitude")
53     ax1.grid(True)
54
55     ax2.clear()
56     ax2.plot(sinyal_rppg[-100:], color='green')
57     ax2.set_title("Sinyal rPPG")
58     ax2.set_xlabel("Waktu")
59     ax2.set_ylabel("Amplitude")
60     ax2.grid(True)
61
62     # Refresh tampilan matplotlib
63     plt.pause(0.01)
64
65     # Keluar dari loop jika tombol 'Esc' ditekan
66     if cv2.waitKey(1) & 0xFF == 27:
67         break
68
69     # Membersihkan sumber daya
70     cap.release()
71     cv2.destroyAllWindows()
72
73 if __name__ == "__main__":
74     main()
75

```

Kode 1: main.py

2.2 respiration.py

File ini berisi modul untuk mengekstraksi sinyal respirasi dari sebuah frame video. Parameter yang digunakan adalah frame video sebagai input, dan akan mengembalikan nilai sinyal respirasi yang diekstraksi. Berikut isi dari file *respiration.py*

```

1
2     import numpy as np
3     def ekstrak_sinyal_respirasi(frame):
4         # Menggunakan intensitas rata-rata piksel
5         nilai_sinyal = np.mean(frame)
6         return nilai_sinyal
7

```

Kode 2: respiration.py

2.3 rppg.py

File ini berisi modul untuk mengekstraksi sinyal remote-photoplethysmography (rPPG) dari sebuah frame video. Parameter yang digunakan adalah frame video sebagai input, dan akan mengembalikan nilai sinyal rPPG yang diekstraksi. Berikut isi dari file *rppg.py*

```
1 import numpy as np
2 def ekstrak_sinyal_rppg(frame):
3     # Menggunakan intensitas rata-rata piksel
4     nilai_sinyal = np.std(frame)
5     return nilai_sinyal
6
7
```

Kode 3: rppg.py

2.4 utils.py

File ini berisi fungsi-fungsi yang akan digunakan dalam pemrosesan sinyal. Pada percobaan ini, saya mencoba menerapkan 2 filter yaitu *butterworth filter* dan *bandpass filter*. Berikut isi dari file *utils.py*

```
1 import numpy as np
2 from scipy.signal import butter, lfilter, filtfilt
3
4 def bandpass_filter(data, lowcut, highcut, fs, order=4):
5     if len(data) == 0: # Jika data kosong
6         return np.array(data) # Output array([], dtype=float64)
7
8     nyquist = 0.5 * fs # Frekuensi Nyquist
9     low = lowcut / nyquist
10    high = highcut / nyquist
11    b, a = butter(order, [low, high], btype='band')
12
13    # Pastikan panjang data lebih besar dari padlen
14    padlen = 3 * max(len(b), len(a))
15    if len(data) <= padlen: # Jika data terlalu pendek
16        return np.array(data) # Kembalikan data apa adanya
17    return filtfilt(b, a, data)
18
19 def butterworth_filter(data, cutoff, fs, order=4, filter_type='low'):
20     nyquist = 0.5 * fs # Frekuensi Nyquist
21     normalized_cutoff = cutoff / nyquist # Normalisasi cutoff
22     b, a = butter(order, normalized_cutoff, btype=filter_type, analog=False)
23     return lfilter(b, a, data)
24
```

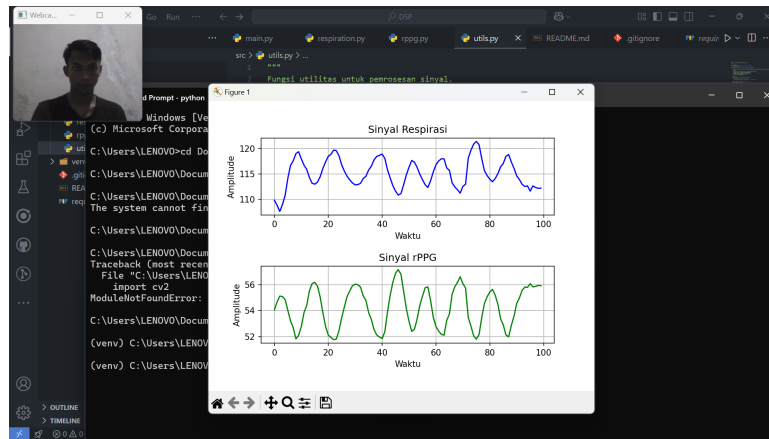
Kode 4: utils.py

3 Percobaan

Setelah implementasi kode dilakukan, barulah melakukan percobaan dengan menjalankan program yang sudah dibuat. Untuk menjalankan program ini, kita hanya perlu membuka terminal dan memasukkan perintah *"python src/main.py"*

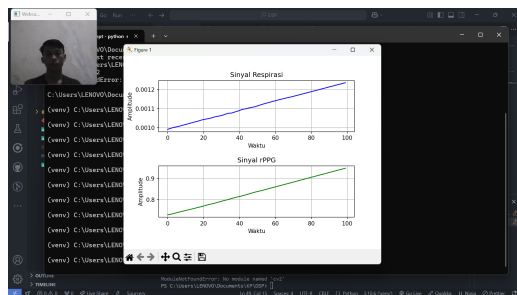
3.1 Hasil Percobaan

Percobaan pertama dilakukan tanpa filtering dan berikut hasilnya.

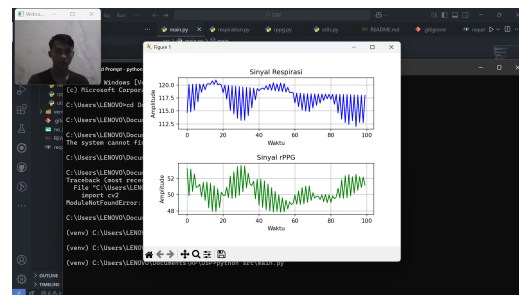


Gambar 2: Percobaan Tanpa Filter

Setelah itu, saya mencoba menerapkan butterworth filter dengan cutoff frekuensi untuk respirasi 0,3 Hz dan cutoff frekuensi untuk rPPG 1,5 Hz. Tidak hanya itu, saya juga mencoba menerapkan bandpass filter dengan lowcut = 0.1 Hz, highcut = 0.5 Hz untuk respirasi, dan lowcut = 0.8 Hz, highcut = 2 Hz untuk rPPG. Tetapi kedua percobaan menggunakan filter ini mendapatkan hasil yang gagal dan mohon maaf saya tidak mengerti bagaimana caranya agar berhasil. Berikut adalah hasil kedua percobaan menggunakan filter.



(a) Butterworth Filter



(b) Bandpass Filter

Gambar 3: Percobaan Menggunakan Filter

3.2 Kesimpulan

Dari hasil percobaan yang dilakukan, dapat disimpulkan bahwa metode filtering yang diterapkan, baik menggunakan Butterworth filter maupun Bandpass filter, tidak memberikan hasil yang sesuai dengan ekspektasi. Meskipun filter telah dirancang untuk menyaring frekuensi tertentu sesuai dengan karakteristik sinyal respirasi dan rPPG, hasil yang diperoleh kurang memadai karena tidak sepenuhnya mampu memisahkan noise dari sinyal utama. Faktor lain yang memengaruhi adalah variasi intensitas cahaya lingkungan dan pergerakan objek dalam video, yang menghasilkan artefak tambahan pada sinyal. Selain itu, parameter filter yang digunakan mungkin kurang optimal untuk kondisi video yang bervariasi, sehingga perlu dilakukan penyesuaian lebih lanjut, seperti pemilihan frekuensi cutoff yang lebih adaptif atau penerapan filter spasial dan temporal yang lebih canggih untuk meningkatkan akurasi sinyal yang dihasilkan.

Tetapi pada kenyaaannya, faktor utama kegagalan ini adalah kurangnya pemahaman saya pribadi tentang bagaimana cara penggunaan filter yang tepat. Saya memohon maaf karena tidak mendengarkan penjelasan saat kelas sehingga hal ini bisa terjadi. Sekian dan terimakasih.