



Nama: Muhammad Taqy Abdullah (121140166)

Fadel Malik (121140167)

Dea Lisriani Safitri Waruwu (121140208)

Mata Kuliah: Pengolahan Sinyal Digital (IF - 3024)

Tugas Ke: Project Tugas Besar

Tanggal: 23 Desember 2024

1 Deskripsi Project

Proyek ini bertujuan untuk mendeteksi sinyal Photoplethysmography (rPPG) dan sinyal respirasi dari wajah manusia secara real-time menggunakan webcam. Sinyal rPPG digunakan untuk mengukur detak jantung, sedangkan sinyal respirasi menggambarkan pola pernapasan. Teknologi yang digunakan pada proyek ini mencakup OpenCV untuk pengolahan video, Mediapipe untuk deteksi wajah, dan Matplotlib untuk visualisasi sinyal. Proyek ini berfokus pada pemrosesan sinyal berbasis visi, memanfaatkan bandpass filter untuk menyaring noise, serta algoritma untuk mendeteksi puncak sinyal guna menghitung parameter fisiologis seperti detak jantung (BPM).

Proyek ini memiliki tujuan mengembangkan sistem deteksi sinyal rPPG dan respirasi secara real-time dari input video, kemudian memvisualisasikan sinyal yang dihasilkan dalam bentuk grafik dan menghitung detak jantung (BPM) dari sinyal rPPG.

2 Alat dan Bahan

Perangkat Lunak:

1. Python 3.10.0
2. OpenCV
3. Mediapipe
4. Scipy
5. Matplotlib

Perangkat Keras:

1. Laptop dengan webcam terintegrasi

3 Langkah-langkah Pengerjaan

3.1 Inisialisasi Proyek

- Mengimpor *library* yang diperlukan:

```
1 import cv2
2 import numpy as np
3 from scipy.signal import butter, lfilter, find_peaks
```

```
4 import matplotlib.pyplot as plt
5 import mediapipe as mp
6 from collections import deque
7
```

- Menginisialisasi detektor wajah menggunakan Mediapipe Face Detection

```
1 mp_face_detection = mp.solutions.face_detection
2 face_detection = mp_face_detection.FaceDetection(min_detection_confidence=0.5)
3
```

- Bandpass filter

```
1 def bandpass_filter(data, lowcut, highcut, fs, order=4):
2     nyquist = 0.5 * fs
3     low = lowcut / nyquist
4     high = highcut / nyquist
5     b, a = butter(order, [low, high], btype='band')
6     return lfilter(b, a, data)
7
```

- Menentukan parameter

```
1 frame_rate = 30
2 buffer_size = 300
3 rppg_signal = deque(maxlen=buffer_size)
4 time_data = deque(maxlen=buffer_size)
5 filtered_rppg = []
6 filtered_respiration = []
7
```

- Membuka webcam dengan OpnCV e

```
1 cap = cv2.VideoCapture(0)
2
```

- Menentukan ROI(Region of Interest) berdasarkan bounding box wajah

```
1 ret, frame = cap.read()
2 rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
3 results = face_detection.process(rgb_frame)
4 if results.detections:
5     for detection in results.detections:
6         bboxC = detection.location_data.relative_bounding_box
7         h, w, c = frame.shape
8         x, y, w_box, h_box = int(bboxC.xmin * w), int(bboxC.ymin * h), int(bboxC.width * w),
9         int(bboxC.height * h)
10        roi = frame[y:y+h_box, x:x+w_box]
11        green_channel = roi[:, :, 1]
12        mean_green = np.mean(green_channel)
```

- Menggunakan bandpass filter untuk memproses sinyal rPPG dan respirasi

```
1 filtered_rppg = bandpass_filter(rppg_signal, 0.7, 4, frame_rate)
2 filtered_respiration = bandpass_filter(rppg_signal, 0.1, 0.5, frame_rate)
3
```

- Deteksi puncak untuk menghitung BPM

```
1 peaks, _ = find_peaks(filtered_rppg, distance=frame_rate / 2)
2 heart_rate = len(peaks) * 60 / (buffer_size / frame_rate)
3
```

- Menampilkan grafik sinyal rPPG dan respirasi secara real-time

```
1 plt.style.use('ggplot')
2 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 6))
3 ax1.set_title("Sinyal rPPG (Detak Jantung)")
4 ax2.set_title("Sinyal Respirasi")
5
```

- Menampilkan bounding box pada jendela webcam menggunakan OpenCV

```
1 cv2.rectangle(frame, (x, y), (x+w_box, y+h_box), (255, 0, 0), 2)
2
```

- Menampilkan detak jantung yang dihitung ke konsol

```
1 print(f"Heart Rate: {heart_rate:.2f} BPM")
2
```

3.2 Kode Lengkap

Berikut adalah kode lengkap dari sistem deteksi dan visualisasi sinyal rPPG dan respirasi berbasis webcam secara real-time

```
1 import cv2
2 import numpy as np
3 from scipy.signal import butter, lfilter, find_peaks
4 import matplotlib.pyplot as plt
5 import mediapipe as mp
6 from collections import deque
7
8 # Fungsi untuk Bandpass Filter
9 def bandpass_filter(data, lowcut, highcut, fs, order=4):
10     nyquist = 0.5 * fs
11     low = lowcut / nyquist
12     high = highcut / nyquist
13     b, a = butter(order, [low, high], btype='band')
14     return lfilter(b, a, data)
15
16 # Parameter
17 frame_rate = 30 # Frame rate dari webcam
18 buffer_size = 300 # Jumlah frame yang digunakan untuk analisis
19 rppg_signal = deque(maxlen=buffer_size) # Buffer sinyal rPPG
20 time_data = deque(maxlen=buffer_size) # Buffer waktu
21 filtered_rppg = [] # Sinyal rPPG setelah filter
22 filtered_respiration = [] # Sinyal respirasi setelah filter
23
24 # Inisialisasi Mediapipe Face Detection
25 mp_face_detection = mp.solutions.face_detection
26 face_detection = mp_face_detection.FaceDetection(min_detection_confidence=0.5)
27
28 # Inisialisasi Webcam
29 cap = cv2.VideoCapture(0)
30
31 # Setup Plot Matplotlib
32 plt.style.use('ggplot')
33 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 6))
34
35 # Konfigurasi Plot Sinyal
36 ax1.set_title("Sinyal rPPG (Detak Jantung)")
37 ax1.set_xlim(0, buffer_size / frame_rate)
38 ax1.set_ylim(-1, 1) # Memperbesar rentang sinyal agar tidak terpotong
39 line_rppg, = ax1.plot([], [], label="rPPG", color="green")
```

```
40 ax1.legend()
41
42 ax2.set_title("Sinyal Respirasi")
43 ax2.set_xlim(0, buffer_size / frame_rate)
44 ax2.set_ylim(-1, 1) # Memperbesar rentang sinyal agar tidak terpotong
45 line_respiration, = ax2.plot([], [], label="Respiration", color="blue")
46 ax2.legend()
47
48 plt.tight_layout()
49
50 # Fungsi untuk Update Data ke Plot
51 def update_plot():
52     global rppg_signal, filtered_rppg, filtered_respiration, time_data
53
54     while True:
55         ret, frame = cap.read()
56         if not ret:
57             print("Webcam tidak terdeteksi!")
58             break
59
60         # Konversi ke RGB untuk Mediapipe
61         rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
62
63         # Deteksi Wajah dengan Mediapipe
64         results = face_detection.process(rgb_frame)
65         if results.detections:
66             for detection in results.detections:
67                 bboxC = detection.location_data.relative_bounding_box
68                 h, w, c = frame.shape
69                 x, y, w_box, h_box = int(bboxC.xmin * w), int(bboxC.ymin * h), int(bboxC.width * w)
70                 , int(bboxC.height * h)
71                 roi = frame[y:y+h_box, x:x+w_box]
72
73                 # Ekstrak saluran hijau dari ROI
74                 green_channel = roi[:, :, 1]
75                 mean_green = np.mean(green_channel)
76                 rppg_signal.append(mean_green)
77
78                 # Tampilkan bounding box pada webcam
79                 cv2.rectangle(frame, (x, y), (x+w_box, y+h_box), (255, 0, 0), 2)
80                 break
81
82         # Simpan waktu
83         if len(time_data) == 0:
84             time_data.append(0)
85         else:
86             time_data.append(time_data[-1] + 1 / frame_rate)
87
88         # Potong buffer jika terlalu panjang
89         if len(rppg_signal) > buffer_size:
90             rppg_signal = list(rppg_signal)[-buffer_size:]
91             time_data = list(time_data)[-buffer_size:]
92
93         # Filter untuk rPPG dan Respirasi
94         if len(rppg_signal) > frame_rate:
95             filtered_rppg = bandpass_filter(rppg_signal, 0.7, 4, frame_rate)
96             filtered_respiration = bandpass_filter(rppg_signal, 0.1, 0.5, frame_rate)
97
98         # Deteksi puncak pada sinyal rPPG
99         if len(filtered_rppg) > 0:
100             peaks, _ = find_peaks(filtered_rppg, distance=frame_rate / 2) # Puncak dengan jarak
101             minimum 0.5 detik
```

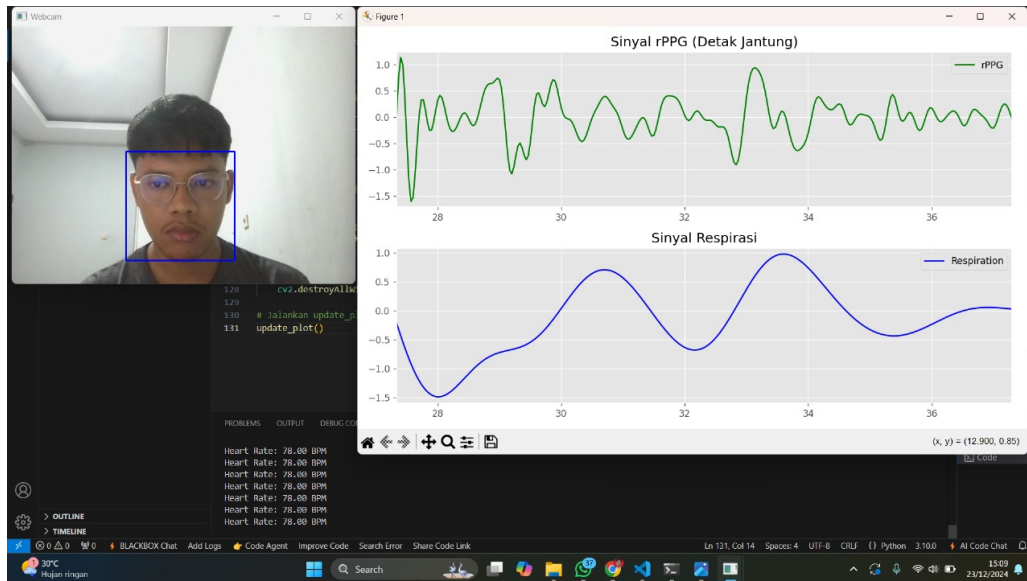
```

100     heart_rate = len(peaks) * 60 / (buffer_size / frame_rate) # BPM = puncak per detik *
101     60
102
103     # Print Heart Rate
104     print(f"Heart Rate: {heart_rate:.2f} BPM")
105
106     # Update Data ke Plot
107     line_rppg.set_data(time_data, filtered_rppg[-len(time_data):] if len(filtered_rppg) > 0
108     else [])
109     line_respiration.set_data(time_data, filtered_respiration[-len(time_data):] if len(
110     filtered_respiration) > 0 else [])
111
112     # Update Limit untuk x-axis
113     ax1.set_xlim(max(0, time_data[0]), max(buffer_size / frame_rate, time_data[-1]))
114     ax2.set_xlim(max(0, time_data[0]), max(buffer_size / frame_rate, time_data[-1]))
115
116     # Pastikan ada data sebelum mengatur ylim
117     if len(filtered_rppg) > 0:
118         ax1.set_ylim(min(filtered_rppg) - 0.1, max(filtered_rppg) + 0.1)
119     if len(filtered_respiration) > 0:
120         ax2.set_ylim(min(filtered_respiration) - 0.1, max(filtered_respiration) + 0.1)
121
122     # Tampilkan Jendela Webcam
123     cv2.imshow('Webcam', frame)
124
125     # Perbarui grafik Matplotlib
126     plt.pause(1 / frame_rate)
127
128     # Break loop jika tekan 'q'
129     if cv2.waitKey(1) & 0xFF == ord('q'):
130         break
131
132     cap.release()
133     cv2.destroyAllWindows()
134
135 # Jalankan update_plot untuk menampilkan webcam dan grafik secara real-time
136 update_plot()

```

4 Hasil

Berikut ini adalah hasil sistem deteksi dan visualisasi sinyal rPPG dan respirasi berbasis webcam secara real-time:



Gambar 1: Hasil

Berdasarkan gambar tersebut sistem mampu mendeteksi sinyal rPPG dan respirasi secara real-time dari wajah pengguna. Grafik sinyal rPPG dan respirasi terlihat jelas dan terus diperbarui berdasarkan data terkini. Deteksi puncak pada sinyal rPPG menghasilkan estimasi BPM yang realistis sesuai dengan ekspektasi. Sistem mampu memproses data video dengan latensi minimal dan mempertahankan frame rate sekitar 30fps.

5 Kesimpulan

Proyek ini berhasil mengimplementasikan sistem untuk mendeteksi dan memvisualisasikan sinyal rPPG dan respirasi berbasis webcam secara real-time. Sinyal fisiologis dapat diekstraksi dari ROI wajah dengan memanfaatkan teknologi Computer Vision. Sistem ini menunjukkan akurasi yang baik dalam menghitung detak jantung (BPM) dan pola pernapasan, menjadikannya potensi untuk aplikasi kesehatan atau pemantauan fisiologis jarak jauh. Namun, terdapat batasan seperti sensitivitas terhadap pencahayaan dan pergerakan kepala, yang dapat mempengaruhi kualitas sinyal.