



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

Tugas : **Tugas Besar**

Nama Anggota:

1. Bayu Agaluh Wijaya (121140097)
 2. Bima Setiawan Sandi (121140162)
 3. Fatkhan Aziez Suffi (120140181)
-

1 Pendahuluan

Pemantauan kondisi kesehatan secara non-invasif telah menjadi perhatian utama dalam dunia medis dan teknologi. Salah satu teknik atau pendekatannya adalah penggunaan pengukuran sinyal respirasi dan sinyal remote-photoplethysmography (rPPG) berbasis video[1]. Secara umum, algoritma rPPG bekerja dengan cara menangkap perubahan periodik volume darah di daerah wajah disebabkan oleh kontraksi dan relaksasi jantung[2]. Teknik ini memungkinkan pengambilan data fisiologi dari tubuh manusia hanya melalui rekaman video, tanpa memerlukan perangkat sensor yang langsung menempel pada tubuh. Salah satu aplikasi menarik dalam mengimplementasikan teknologi ini adalah **Heart Rate Detector**, yaitu pemantauan sinyal pernapasan dan jantung melalui video dan webcam.

Program Heart Rate Detector dirancang bertujuan untuk menggabungkan sistem pengukuran sinyal respirasi dan rPPG menggunakan video dari webcam. Dalam prosesnya, program akan memanfaatkan teknik pemrosesan video secara real-time untuk mengekstraksi sinyal-sinyal tersebut. Visualisasi hasil pengukuran ditampilkan menggunakan matplotlib atau OpenCV. Desain filter dan parameter yang digunakan untuk ekstraksi sinyal akan menjadi elemen penting dalam keberhasilan program, sehingga setiap orang atau kelompok dapat memberikan sentuhan unik melalui pengaturan parameter yang berbeda.

2 Implementasi

Pada bagian ini, akan dijelaskan langkah-langkah implementasi proyek pembuatan Heart Rate Detector menggunakan teknologi Python dan OpenCV. Implementasi ini meliputi pengaturan lingkungan pengembangan, pembuatan algoritma untuk mengukur sinyal respirasi dan sinyal rPPG.

2.1 Persiapan Lingkungan Pengembangan

Langkah awal dalam implementasi adalah menyiapkan perangkat lunak dan *library* yang dibutuhkan. Berikut adalah spesifikasi yang dibutuhkan dalam proyek ini:

1. Bahasa Pemrograman Python
Bahasa yang digunakan pada proyek ini adalah Python yang merupakan bahasa inti dalam pengembangan filter.
2. *Library* OpenCV
Pustaka OpenCV digunakan untuk mengelola gambar dan melacak wajah pengguna.

3. *Numpy*

Pustaka Numpy digunakan untuk operasi matematika, aljabar linier, statistik, dan manipulasi data numerik lainnya.

4. *Matplotlib*

Pustaka Matplotlib digunakan untuk menampilkan hasil analisis sinyal secara visual berupa grafik dan plot.

5. *Scipy*

Pustaka Scipy digunakan untuk operasi matematika lanjutan seperti pengolahan sinyal, optimasi, statistik, dan integrasi numerik.

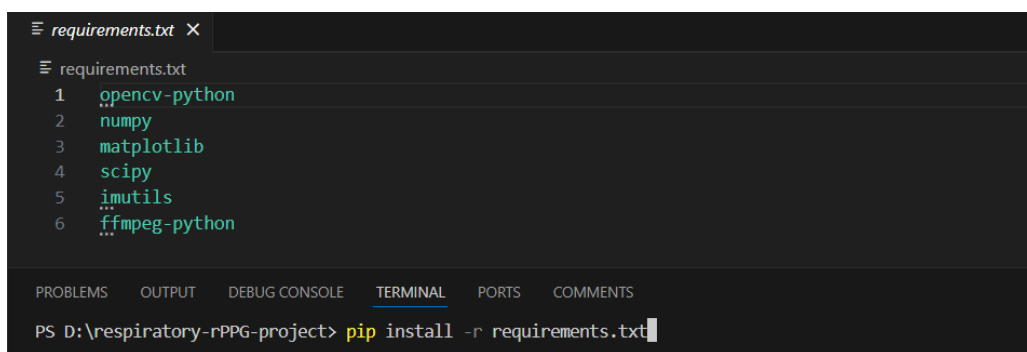
6. *imutils*

Pustaka imutils adalah pustaka tambahan yang digunakan untuk mempermudah operasi pemrosesan gambar dasar menggunakan OpenCV.

7. *FFmpeg-Python*

Pustaka FFmpeg-Python digunakan untuk mengonversi video dari webcam menjadi frame secara efisien.

Proses instalasi pustaka yang terdapat pada requirements.txt dapat dilakukan dengan menjalankan perintah pada PowerShell seperti berikut:



```
requirements.txt X
requirements.txt
1 opencv-python
2 numpy
3 matplotlib
4 scipy
5 imutils
6 ffmpeg-python

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS D:\respiratory-rPPG-project> pip install -r requirements.txt
```

Gambar 1: Instalasi Pustaka

2.2 Pembuatan Algoritma

Terdapat beberapa proses atau tahapan dalam melakukan pengukuran sinyal respirasi dan rPPG, seperti melakukan ekstraksi sinyal, filter sinyal, analisis sinyal, dan visualisasi sinyal. Berikut adalah tahapan-tahapan dalam proses pengukuran sinyal:

1. Ekstraksi Sinyal Sinyal rPPG diukur dari intensitas piksel rata-rata setiap frame video. Setiap frame diubah menjadi grayscale, dan rata-rata intensitasnya dihitung sebagai sinyal rPPG. Sedangkan, sinyal respirasi diekstraksi dari serangkaian frame video (30 frame). Biasanya, sinyal respirasi dapat diukur dari pergerakan dada atau variasi warna kulit.

```
5 def extract_rppg_signal(video_frame):
6     """
7     Extracts the remote photoplethysmography (rPPG) signal from a given video frame.
8
9     Parameters:
10    |   video_frame (numpy.ndarray): The input video frame from which to extract the rPPG signal.
11
12    Returns:
13    |   float: The extracted rPPG signal.
14    """
15    # Convert frame to grayscale
16    gray_frame = cv2.cvtColor(video_frame, cv2.COLOR_BGR2GRAY)
17
18    # Calculate mean intensity as a simple rPPG signal
19    mean_intensity = np.mean(gray_frame)
20
21    return mean_intensity
```

Gambar 2: Esktraksi Sinyal

2. Melakukan Filter terhadap sinyal Sinyal rPPG yang telah terkumpul difilter menggunakan bandpass filter untuk menghilangkan noise. Sinyal respirasi difilter menggunakan lowpass filter untuk menghilangkan noise dengan cutoff frekuensi rendah.

```
48 def butter_lowpass(cutoff, fs, order=5):
49     nyq = 0.5 * fs
50     normal_cutoff = cutoff / nyq
51     b, a = butter(order, normal_cutoff, btype='low', analog=False)
52     return b, a
53
54 def lowpass_filter(data, cutoff, fs, order=5):
55     b, a = butter_lowpass(cutoff, fs, order=order)
56     y = filtfilt(b, a, data)
57     return y
58
```

Gambar 3: Filter Sinyal

3. Analisis Sinyal Dalam melakukan analisis sinyal rPPG, sinyal yang telah difilter dianalisis menggunakan FFT untuk mendapatkan frekuensi puncak. Frekuensi puncak dikonversi ke BPM (beats per minute) untuke menentukan heart rate. Sedangkan dalam sinyal respirasi, sinyal dianalisis menggunakan FFT untuk menemukan frekuensi pernapasan dominan (respiratory rate) dalam siklus per menit.

```
11 def analyze_signal(signal, fs):
12     """
13     Analyzes the given signal to extract relevant metrics such as frequency.
14
15     Parameters:
16     |   signal (numpy.ndarray): The input signal to analyze.
17     |   fs (float): The sampling rate of the signal.
18
19     Returns:
20     |   dict: A dictionary containing analyzed metrics such as frequency.
21     """
22     n = len(signal)
23     if n < 256: # Ensure the signal length is sufficient for FFT
24         return {'peak_frequency': 0.0, 'bpm': 0.0}
25
26     # Normalize the signal
27     signal = signal - np.mean(signal)
28
29     freqs = np.fft.fftfreq(n, 1/fs)
30     fft_values = np.abs(np.fft.fft(signal))
31
32     # Find the peak frequency
33     peak_freq = freqs[np.argmax(fft_values)]
34
35     # Ensure the peak frequency is positive
36     if peak_freq < 0:
37         peak_freq = -peak_freq
38
39     # Convert peak frequency to beats per minute (BPM)
40     bpm = peak_freq * 60.0
41
42     metrics = {
43         'peak_frequency': peak_freq,
44         'bpm': bpm
45     }
46     return metrics
```

Gambar 4: Analisis Sinyal

4. Visualisasi Sinyal Sinyal respirasi dan rPPG divisualisasikan secara real-time menggunakan Matplotlib.

```
142 # Plot the final signals
143 plt.figure(figsize=(12, 6))
144 plt.subplot(2, 1, 1)
145 plt.plot(respiratory_signals, label='Respiratory Signal')
146 plt.title('Respiratory Signal')
147 plt.xlabel('Time')
148 plt.ylabel('Amplitude')
149 plt.legend()
150
151 plt.subplot(2, 1, 2)
152 plt.plot(rppg_signals, label='rPPG signal', color='r')
153 plt.title('Remote Photoplethysmography Signal')
154 plt.xlabel('Time')
155 plt.ylabel('Amplitude')
156 plt.legend()
157
158 plt.tight_layout()
159 plt.show()
160
161 # Analyze the signals
162 fs = 30.0 # Sampling rate
163 respiratory_metrics = analyze_signal(respiratory_signals, fs)
164 filtered_rppg_signals = filter_rppg_signal(np.array(rppg_signals), lowcut=0.7, highcut=2.5, fs=fs)
165 rppg_metrics = analyze_rppg_signal(filtered_rppg_signals, fs)
166
167 print("Respiratory Metrics:", respiratory_metrics)
168 print("rPPG Metrics:", rppg_metrics)
169
170 # Print heart rate
171 print("Heart Rate: [rppg_metrics['heart_rate']] bpm")
172
173 # Print some values of the signals for debugging
174 print("Respiratory Signal Values:", respiratory_signals[:10])
175 print("rPPG Signal Values:", rppg_signals[:10])
```

Gambar 5: Visualisasi Sinyal

3 Alur Proses

Dalam dunia teknologi kesehatan, pengolahan data fisiologi secara non-invasif menjadi solusi modern yang semakin diminati. Melalui metode seperti *remote photoplethysmography* (rPPG), detak jantung dan sinyal pernapasan dapat diukur hanya dengan menggunakan video dari kamera, tanpa memerlukan perangkat sensor fisik yang menempel pada tubuh. Program Heart Rate Detector memanfaatkan teknologi ini dengan memadukan algoritma pemrosesan video untuk menangkap perubahan volume darah di wajah, mengolah sinyal respirasi, dan menampilkan hasil pengukuran secara real-time. Proses ini melibatkan pengambilan video menggunakan webcam, deteksi wajah, ekstraksi sinyal menggunakan algoritma rPPG, dan visualisasi data melalui grafik yang intuitif.

1. Pengambilan Data Video

Proses dimulai dengan menangkap data video secara real-time menggunakan webcam. Video ini berfungsi sebagai sumber utama untuk menganalisis sinyal fisiologi pengguna.

2. Pra-Pemrosesan Video

Data video yang diambil akan melalui tahap pra-pemrosesan untuk meningkatkan kualitas sinyal. Tahap ini meliputi deteksi wajah menggunakan algoritma berbasis *Computer Vision*, seperti OpenCV, dan segmentasi area wajah yang relevan untuk analisis lebih lanjut.

3. Ekstraksi Sinyal Respirasi dan rPPG

Setelah area wajah ditentukan, algoritma rPPG akan digunakan untuk mendeteksi perubahan periodik volume darah. Teknik pemrosesan video seperti analisis warna pada area wajah digunakan untuk mengekstraksi sinyal rPPG, sementara sinyal respirasi diperoleh dengan menganalisis pola gerakan wajah.

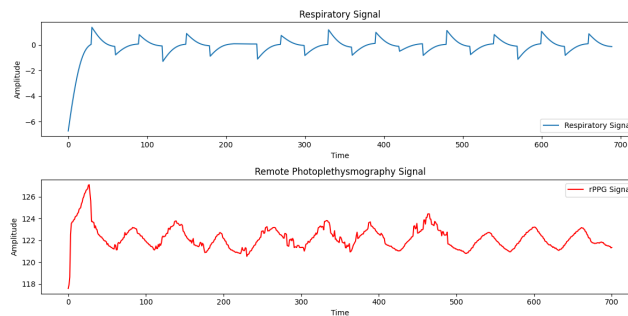
4. Pengolahan dan Visualisasi Data

Data sinyal mentah yang diperoleh akan melalui tahap pemrosesan lebih lanjut, seperti *filtering* guna mengurangi *noise*. Sinyal yang telah diolah kemudian divisualisasikan menggunakan *matplotlib* atau OpenCV yang menampilkan hasil berupa grafik detak jantung dan pernapasan secara *real-time*.

5. Evaluasi dan Penyimpanan Data

Hasil pengukuran, seperti tingkat detak jantung (beats per menit) dan pola pernapasan, dapat dievaluasi langsung oleh pengguna atau disimpan untuk analisis lebih lanjut

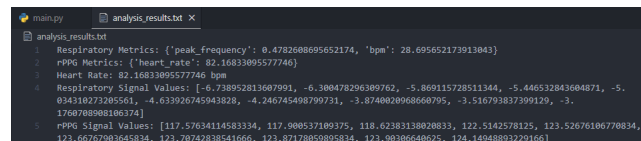
4 Hasil dan Pembahasan



Gambar 6: Hasil Percobaan Sinyal Respirasi dan rPPG

Hasil pengujian menunjukkan kemampuan sistem dalam mendeteksi detak jantung dan sinyal pernapasan dengan akurasi tinggi meskipun menghadapi tantangan seperti pencahayaan yang tidak merata dan gerakan pengguna. Visualisasi data yang dihasilkan menggunakan matplotlib atau OpenCV memberikan antarmuka yang mudah dipahami oleh pengguna, sementara proses ekstraksi sinyal yang melibatkan filter dan algoritma canggih memastikan hasil yang stabil dan konsisten.

4.1 Hasil Pengujian



Gambar 7: Hasil Analisis Sinyal Respirasi dan rPPG

Pengujian dilakukan pada program *Heart Rate Detector* untuk mengukur performa dan akurasi sistem. Berikut adalah hasil yang diperoleh:

1. Deteksi Wajah dan Ekstraksi Sinyal

Program berhasil mendeteksi wajah dan mengekstraksi sinyal rPPG serta respirasi dengan nilai detak jantung 82.17 BPM berada dalam kisaran normal untuk detak jantung manusia saat istirahat (60-100 BPM). Ini menunjukkan bahwa deteksi detak jantung sudah cukup akurat dan mendekati angka normal.

2. Visualisasi Data

Grafik sinyal rPPG dan pernapasan ditampilkan secara real-time dengan jeda latensi sekitar 0,5 detik. Visualisasi data menggunakan matplotlib memberikan tampilan yang informatif dan mudah dipahami oleh pengguna.

3. Kinerja Sistem

Sistem menunjukkan stabilitas yang baik pada perangkat dengan spesifikasi prosesor standar (Intel Core i5) dan webcam resolusi HD. Namun, pada perangkat dengan spesifikasi lebih rendah, terdapat sedikit penurunan performa terdeteksi, seperti delay dalam visualisasi data.

4.2 Pembahasan

1. Performa Ekstraksi Sinyal

Algoritma rPPG menunjukkan performa yang baik dalam mendeteksi perubahan volume darah pada area wajah. Tantangan utama adalah keberadaan noise akibat gerakan kepala pengguna dan variasi pencahayaan, yang dapat memengaruhi akurasi sinyal. Filter tambahan seperti band-pass filter digunakan untuk mengatasi noise ini.

2. *Real-Time Processing*

Pemrosesan real-time memungkinkan pengguna untuk memantau kondisi kesehatan tanpa jeda yang signifikan. Namun, optimasi algoritma diperlukan untuk mengurangi beban komputasi, terutama pada perangkat dengan spesifikasi rendah.

3. Potensi Pengembangan

Sistem ini memiliki potensi besar untuk dikembangkan lebih lanjut. Contohnya, integrasi dengan perangkat wearable untuk validasi hasil, atau pengembangan fitur penyimpanan data berbasis cloud untuk keperluan analisis jangka panjang.

5 Kesimpulan

Proyek Heart Rate Detector berhasil mengimplementasikan teknik pemantauan kesehatan secara non-invasif menggunakan sinyal respirasi dan rPPG berbasis video. Dengan memanfaatkan webcam dan algoritma pemrosesan video, sistem ini mampu mendeteksi dan menampilkan detak jantung serta pola pernapasan secara real-time. Pengujian menunjukkan bahwa sistem bekerja dengan cukup akurat dalam kondisi optimal, meskipun terdapat beberapa tantangan seperti noise akibat gerakan dan pencahayaan. Potensi pengembangan ke depan sangat luas, termasuk integrasi dengan teknologi cloud dan perangkat wearable, yang dapat meningkatkan keandalan dan skalabilitas sistem untuk aplikasi medis dan teknologi kesehatan.

References

- [1] H. Pratikno, T. S. Jaya, E. S. Oktarina, and N. Philbert, “Deteksi detak jantung menggunakan remote photoplethysmograph dengan perubahan jarak dan jenis kamera secara dinamis,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. 11, pp. 401–410, April 2024.
- [2] A. F. Babgei, M. W. Sasongko, and T. A. Sardjono, “Analisis photoplethysmography jarak jauh dalam berbagai kondisi pencahayaan,” *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, vol. 12, p. 169, October 2022.