



---

Nama: **Ericson Chandra Sihombing (121450026), Shula Talitha Ardhya Putri (121450087)**  
Tugas Ke: **Final Project IF3024**  
Mata Kuliah: **Digital Signal Processing (IF3024)** Tanggal: 31/05/2025

---

## 1 Ketentuan Proyek:

Proyek ini meminta pembuatan program yang menggabungkan sistem pengukuran sinyal pernapasan dan sistem pengukuran remote-photoplethysmography (rPPG). Program ini akan menerima input video dari webcam dan memprosesnya secara real-time untuk menampilkan sinyal pernapasan dan sinyal rPPG. Program tidak wajib memiliki graphical user interface (GUI), visualisasi dapat menggunakan Matplotlib atau CV2, namun pembuatan GUI akan mendapatkan nilai tambah. Perbedaan antar kelompok atau individu terletak pada desain filter dan pemrosesan sinyal, termasuk penentuan parameter untuk ekstraksi sinyal pernapasan dan rPPG.

## 2 Deskripsi Proyek

### 2.1 Latar Belakang

Proyek ini merupakan bagian dari mata kuliah Pengolahan Sinyal Digital (IF3024) di Institut Teknologi Sumatera. Ini adalah upaya untuk mengembangkan sistem pemantauan kesehatan non-invasif yang dapat mengekstrak sinyal pernapasan dan detak jantung (rPPG) menggunakan *webcam*, berdasarkan praktik sebelumnya.

### 2.2 Tujuan

1. **Integrasi Sistem:** Menggabungkan sistem pengukuran sinyal respirasi dan rPPG dari *input* video *webcam*.
2. **Pemrosesan *Real-time*:** Memproses dan menampilkan sinyal pernapasan dan rPPG secara *real-time*.
3. **Optimalisasi Sinyal:** Merancang *filter* dan proses sinyal untuk ekstraksi sinyal pernapasan dan rPPG yang akurat.
4. **Pengembangan GUI:** Menyediakan GUI menggunakan PyQt5 untuk interaksi pengguna yang lebih baik.
5. **Demonstrasi Konsep DSP:** Menunjukkan pemahaman konsep pengolahan sinyal digital dan penerapannya.

### 2.3 Manfaat

1. **Pemantauan Kesehatan Non-invasif:** Memungkinkan pemantauan pernapasan dan detak jantung tanpa kontak fisik.

2. **Edukasi:** Berfungsi sebagai alat pembelajaran praktis tentang pengolahan sinyal digital dan deteksi sinyal biometrik dari video.
3. **Dasar Pengembangan:** Menjadi fondasi untuk aplikasi kesehatan lanjutan, seperti pemantauan jarak jauh.
4. **Fleksibilitas:** Parameter *filter* yang bisa disesuaikan memungkinkan adaptasi terhadap kondisi berbeda.
5. **Aksesibilitas:** Sistem mudah diimplementasikan hanya dengan *webcam* standar.

### 3 Metode Pengembangan Sistem Proyek

Metode yang digunakan dalam aplikasi ini mengandalkan pengambilan video real-time dari webcam menggunakan OpenCV dan deteksi pose tubuh serta wajah dengan MediaPipe. Posisi bahu kiri dan kanan digunakan untuk mengukur gerakan dada sebagai sinyal pernapasan, sementara area wajah di tengah frame (ROI) dimonitor untuk ekstraksi sinyal detak jantung (rPPG) berdasarkan perubahan warna kulit.

Sinyal pernapasan dan rPPG yang diperoleh kemudian diproses menggunakan filter bandpass Butterworth untuk menghilangkan noise dan menjaga frekuensi sinyal yang relevan. Parameter filter dapat disesuaikan oleh pengguna melalui antarmuka GUI. Puncak-puncak sinyal yang sudah difilter dideteksi untuk menghitung frekuensi napas dan detak jantung dalam satuan beats per minute (BPM).

Proses pengambilan video dan pemrosesan sinyal dilakukan pada thread terpisah agar antarmuka pengguna tetap responsif. GUI yang dibangun dengan PyQt5 menampilkan video webcam dengan overlay ROI, grafik sinyal secara real-time, serta informasi BPM dan status kualitas sinyal. Dengan metode ini, aplikasi dapat memantau pernapasan dan detak jantung secara non-invasif dan interaktif menggunakan webcam biasa.

#### 3.1 Perhitungan Teknis dalam Aplikasi

Aplikasi memproses data video dari webcam untuk mendapatkan sinyal pernapasan dan detak jantung secara real-time.

##### 3.1.1 Pengukuran Gerakan Pernapasan

Posisi vertikal dada dihitung dari rata-rata posisi bahu kiri dan kanan pada setiap frame:

$$y_{\text{dada}}(t) = \frac{y_{\text{bahu kiri}}(t) + y_{\text{bahu kanan}}(t)}{2}$$

di mana  $y_{\text{dada}}(t)$  adalah posisi dada pada waktu  $t$ , dan  $y_{\text{bahu kiri}}(t)$  serta  $y_{\text{bahu kanan}}(t)$  adalah posisi vertikal bahu kiri dan kanan. Sinyal pernapasan diperoleh dari perubahan posisi dada antar frame:

$$\Delta y_{\text{dada}}(t) = y_{\text{dada}}(t) - y_{\text{dada}}(t - 1)$$

yang menggambarkan gerakan naik-turun dada sebagai indikator pernapasan.

##### 3.1.2 Ekstraksi Sinyal rPPG

Sinyal detak jantung diambil dari perubahan warna wajah. Nilai warna merah ( $R$ ), hijau ( $G$ ), dan biru ( $B$ ) dinormalisasi untuk mengurangi efek pencahayaan:

$$S(t) = R(t) + G(t) + B(t) + \varepsilon$$

dengan  $\varepsilon$  nilai kecil untuk menghindari pembagian dengan nol. Nilai normalisasi warna adalah:

$$R_n(t) = \frac{R(t)}{S(t)}, \quad G_n(t) = \frac{G(t)}{S(t)}, \quad B_n(t) = \frac{B(t)}{S(t)}$$

Sinyal rPPG dihitung menggunakan metode CHROM:

$$Y_{\text{chrom}}(t) = 1.5 \times R_n(t) + G_n(t) - 1.5 \times B_n(t)$$

yang merepresentasikan perubahan warna akibat denyut nadi.

### 3.1.3 Filter Bandpass

Untuk menghilangkan noise dan mempertahankan frekuensi sinyal penting, sinyal mentah difilter dengan filter Butterworth:

$$x_{\text{filtered}} = \text{filtfilt}(b, a, x)$$

di mana  $b, a$  adalah koefisien filter yang dihitung berdasarkan frekuensi batas bawah, batas atas, dan orde filter.

### 3.1.4 Perhitungan BPM

Puncak-puncak sinyal yang telah difilter,  $\{p_1, p_2, \dots, p_k\}$ , mewakili satu siklus napas atau denyutan jantung. Rata-rata jarak antar puncak dihitung sebagai:

$$\overline{\Delta p} = \frac{1}{k-1} \sum_{i=1}^{k-1} (p_{i+1} - p_i)$$

Interval waktu antar puncak dalam detik dihitung dengan membagi rata-rata jarak dengan frekuensi sampling  $f_s$ :

$$T = \frac{\overline{\Delta p}}{f_s}$$

BPM (beats per minute) kemudian dihitung sebagai:

$$\text{BPM} = \frac{60}{T} = \frac{60 \times f_s}{\overline{\Delta p}}$$

### 3.1.5 Amplitudo Sinyal

Amplitudo sinyal yang telah difilter dihitung dengan metode root mean square (RMS) untuk mengukur kekuatan sinyal:

$$A = \sqrt{\frac{1}{N} \sum_{t=1}^N x_{\text{filtered}}(t)^2}$$

di mana  $N$  adalah jumlah data dalam sinyal.

## 4 Kode Program Python Proyek

Kode program Python yang dibuat adalah aplikasi GUI (Graphical User Interface) yang menggunakan PyQt5, OpenCV (cv2), NumPy, SciPy, dan MediaPipe untuk memproses video dari webcam secara real-time guna mendeteksi sinyal pernapasan dan rPPG (detak jantung).

## 4.1 Ringkasan Fungsionalitas Program

Program ini berfungsi sebagai monitor *real-time* untuk pernapasan dan detak jantung seseorang menggunakan *webcam*. Pengguna dapat melihat sinyal pernapasan dan detak jantung yang diekstrak, mengatur parameter *filter*, dan memantau kualitas sinyal.

## 4.2 Komponen Utama Kode

### 4.2.1 Parameter Global

- **WEBCAM\_INDEX**: Mengatur *webcam* yang digunakan (*default* 0).
- **FRAME\_WIDTH**, **FRAME\_HEIGHT**: Mengatur resolusi *frame* video.
- **FPS\_DEFAULT**: *Frame per second default*.
- **DEFAULT\_LOWCUT\_RESP**, **DEFAULT\_HIGHCUT\_RESP**, **DEFAULT\_ORDER\_RESP**: Parameter *filter band-pass default* untuk sinyal pernapasan.
- **DEFAULT\_LOWCUT\_PPG**, **DEFAULT\_HIGHCUT\_PPG**, **DEFAULT\_ORDER\_PPG**: Parameter *filter bandpass default* untuk sinyal rPPG (detak jantung).
- **BUFFER\_SIZE\_SEC**: Ukuran *buffer* data sinyal (*default* 10 detik).

### 4.2.2 Fungsi Filter (**butter\_bandpass**, **butter\_bandpass\_filter**)

- Mengimplementasikan *filter bandpass* Butterworth menggunakan pustaka SciPy. *Filter* ini digunakan untuk menghilangkan *noise* dan mengisolasi rentang frekuensi sinyal pernapasan dan detak jantung yang diinginkan.
- **butter\_bandpass**: Menghitung koefisien *filter*.
- **butter\_bandpass\_filter**: Menerapkan *filter* pada data sinyal. Fungsi ini juga menangani kasus di mana data terlalu pendek untuk di-*filter*.

### 4.2.3 Kelas **VideoProcessor** (QThread)

Ini adalah *thread* terpisah yang bertugas memproses video dari *webcam* agar aplikasi GUI tetap responsif.

- **Inisialisasi**: Mengatur *buffer* untuk data waktu, sinyal pernapasan, dan *channel* warna RGB. Juga menginisialisasi MediaPipe FaceMesh dan Pose untuk deteksi wajah dan gerak tubuh.
- **set\_filter\_params**: Memungkinkan GUI untuk memperbarui parameter *filter* secara *real-time*.
- **chrom\_rppg**: Mengimplementasikan algoritma CHROM (*Chrominance-based rPPG*) untuk mengekstraksi sinyal rPPG dari perubahan warna rata-rata *channel* R, G, B pada area wajah yang terdeteksi.
- **run (Metode Utama Thread)**:
  - Membuka *webcam* dan mengatur resolusi *frame*.
  - Secara terus-menerus membaca *frame* video.
  - Membalik *frame* secara horizontal (*mirror effect*).
  - Menggunakan MediaPipe Pose untuk mendeteksi posisi bahu kiri dan kanan untuk memperkirakan pergerakan dada (sinyal pernapasan).

- Menentukan *Region of Interest* (ROI) berupa kotak hijau di tengah *frame* untuk ekstraksi sinyal rPPG.
- Menghitung rata-rata warna RGB di dalam ROI untuk sinyal rPPG.
- Menyimpan data sinyal pernapasan dan rPPG mentah ke dalam *buffer*.
- Menerapkan *filter bandpass* pada sinyal pernapasan dan rPPG untuk mendapatkan sinyal yang lebih bersih.
- Menghitung amplitudo (kekuatan) sinyal pernapasan dan rPPG.
- Mengirimkan *signal* ke GUI untuk memperbarui gambar video, data sinyal, informasi waktu wajah stabil, dan informasi amplitudo.
- Menggambar kotak hijau *overlay* di *frame* video.
- **convert\_cv\_qt**: Mengubah *frame* OpenCV menjadi format QImage agar dapat ditampilkan di QLabel PyQt.
- **stop**: Menghentikan *thread* pemrosesan video.

#### 4.2.4 Kelas HelpDialog (QDialog)

- Jendela dialog terpisah yang menyediakan penjelasan detail mengenai setiap parameter *filter* (*Low Cut*, *High Cut*, *Orde Filter*) untuk sinyal pernapasan dan rPPG.
- Juga menjelaskan pentingnya menjaga wajah stabil di kotak hijau selama 30 detik untuk akurasi pengukuran.

#### 4.2.5 Kelas MainWindow (QMainWindow)

Ini adalah kelas utama yang mengelola antarmuka pengguna (GUI).

- **Inisialisasi**:
  - Mengatur judul dan ukuran jendela (`showMaximized()` untuk *fullscreen*).
  - Mengatur *layout* utama menggunakan `QHBoxLayout` dan `QSplitter` untuk membagi tampilan menjadi dua bagian: kiri (video, kontrol, parameter) dan kanan (grafik sinyal).
- **Bagian Kiri**:
  - **video\_label**: Menampilkan *feed* video dari *webcam*.
  - Tombol **start\_button**, **stop\_button**, **apply\_filter\_button**, **reset\_button** untuk mengontrol alur program.
  - **metrics\_label**: Menampilkan FPS, BPM pernapasan, dan BPM detak jantung.
  - **face\_pos\_label**: Memberikan instruksi posisi wajah dalam ROI.
  - **amplitude\_label**: Menampilkan amplitudo sinyal pernapasan dan detak jantung.
  - **filter\_frame**: Berisi *input* untuk parameter *filter* (Respirasi *Low Cut*, *High Cut*, *Order*; PPG *Low Cut*, *High Cut*, *Order*). Menggunakan `QFormLayout` dan `QScrollArea` agar rapi dan dapat digulir.
  - **stable\_info\_label**: Menampilkan informasi durasi wajah stabil di dalam kotak hijau.
  - **signal\_quality\_label**: Menampilkan status kualitas sinyal setelah 30 detik.
  - **science\_info\_label**: Memberikan informasi sains singkat.
- **Bagian Kanan**:

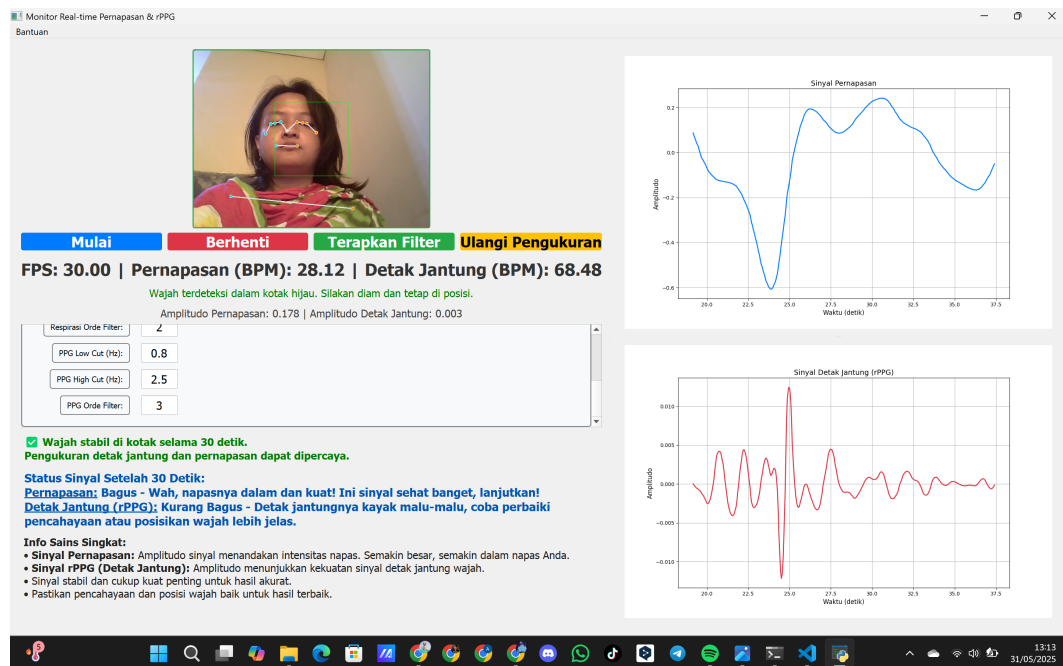
- Dua area *plot* (menggunakan Matplotlib **FigureCanvasQTAgg**) untuk menampilkan grafik sinyal pernapasan (**ax\_resp**) dan sinyal rPPG (**ax\_rppg**) secara *real-time*.
- **Koneksi Sinyal/Slot**: Menghubungkan *signal* dari **VideoProcessor thread** ke *slot* di **MainWindow** untuk memperbarui GUI.
- **create\_menu**: Membuat menu 'Bantuan' untuk mengakses **HelpDialog**.
- **start\_processing**: Memulai *thread VideoProcessor* dan *timer* untuk *update plot*. **stop\_processing**: Menghentikan *thread VideoProcessor* dan *timer plot*.
- **reset\_measurement**: Mereset semua data pengukuran dan label informasi.
- **apply\_filter\_changes**: Membaca nilai dari *input* parameter *filter* di GUI, melakukan validasi, dan mengirimkannya ke **VideoProcessor thread**.
- **update\_image**: Menerima QImage dari *thread* video dan menampilkannya di **video\_label**.
- **store\_data\_for\_plot**: Menerima data sinyal dari *thread* video dan menyimpannya untuk di-*plot*.
- **update\_plots**: Dipanggil secara berkala oleh QTimer untuk memperbarui grafik sinyal. Juga menghitung BPM (napas per menit) dan BPM (detak per menit) dari sinyal yang difilter dengan mendeteksi puncak sinyal.
- **update\_stable\_time\_info**: Memperbarui label yang menampilkan durasi wajah stabil di kotak hijau dan memberikan pesan terkait kualitas pengukuran.
- **update\_amplitude\_info**: Memperbarui label yang menampilkan amplitudo sinyal dan memicu *update signal\_quality\_label* setelah 30 detik stabil.
- **update\_face\_position\_info**: Memberikan umpan balik kepada pengguna mengenai posisi wajah di dalam ROI.
- **update\_science\_info**: Menampilkan informasi ilmiah singkat mengenai interpretasi sinyal.
- **kategorikan\_respirasi, kategorikan\_rppg**: Fungsi untuk mengkategorikan kualitas sinyal pernapasan dan rPPG berdasarkan amplitudo rata-rata sinyal.
- **closeEvent**: Memastikan *thread* video berhenti ketika aplikasi ditutup.

### 4.3 Cara Kerja Singkat

- Pengguna membuka aplikasi dan menekan tombol "Mulai".
- *Thread VideoProcessor* dimulai, mengambil *frame* dari *webcam*.
- Setiap *frame* diproses untuk:
  - Mendeteksi pergerakan dada (menggunakan MediaPipe Pose) untuk sinyal pernapasan.
  - Mengambil rata-rata warna RGB dari area wajah (ROI kotak hijau) untuk sinyal rPPG.
- Sinyal mentah ini kemudian di-*filter* menggunakan *filter bandpass* untuk menghilangkan *noise* dan mengisolasi frekuensi yang relevan.
- Data sinyal yang difilter dan informasi seperti FPS, BPM, dan amplitudo dikirim kembali ke GUI.

- GUI memperbarui tampilan video, grafik sinyal pernapasan dan rPPG secara *real-time*, serta menampilkan informasi metrik dan kualitas sinyal.
- Pengguna dapat menyesuaikan parameter *filter* untuk mencoba mendapatkan sinyal yang lebih baik.

#### 4.4 Visualisasi Sinyal dan Tampilan Program



Gambar 1: Visualisasi Sinyal dan Tampilan Program (Data Terekam)

Visualisasi pada Gambar 1 menunjukkan antarmuka program untuk memantau data real-time, khususnya pernapasan dan detak jantung menggunakan rPPG (remote Photoplethysmography). Berbeda dengan kondisi sebelumnya, gambar ini menunjukkan proses pengukuran yang sedang berjalan dengan data yang sudah terekam.

##### 4.4.1 Bagian Atas (Monitor Real-time Pernapasan rPPG):

1. **Jendela Video Feed di Kiri Atas:** Terlihat wajah subjek sedang terekam. Ini menandakan kamera aktif dan berhasil mendeteksi wajah, yang merupakan langkah krusial untuk pengukuran rPPG.
2. **Tombol-tombol Kontrol:**
  - **Mulai:** Tombol ini menandakan proses pengukuran sudah dimulai.
  - **Berhenti:** Untuk menghentikan proses pengukuran.
  - **Terapkan Filter:** Untuk menerapkan pengaturan filter yang telah ditentukan.
  - **Ulangi Pengukuran:** Untuk memulai ulang pengukuran.
3. **Informasi Status (di bawah tombol kontrol):**
  - **FPS: 30.00:** Menunjukkan *Frames Per Second* (jumlah bingkai per detik) dari video feed adalah 30, yang merupakan kecepatan bingkai yang baik untuk akuisisi data real-time.

- **Pernapasan (BPM): 28.12:** Ini adalah hasil deteksi laju pernapasan dalam *Beats Per Minute*. Nilai 28.12 BPM menunjukkan pernapasan yang relatif cepat atau bisa jadi karena aktivitas fisik ringan atau penyesuaian awal.
  - **Detak Jantung (BPM): 68.48:** Ini adalah hasil deteksi laju detak jantung dalam BPM. Nilai 68.48 BPM berada dalam rentang normal untuk detak jantung istirahat orang dewasa.
  - **Amplitudo Pernapasan: 0.178 | Amplitudo Detak Jantung: 0.003:** Menampilkan amplitudo dari sinyal pernapasan dan detak jantung yang terdeteksi. Amplitudo sinyal pernapasan (0.178) jauh lebih besar daripada sinyal detak jantung (0.003), yang konsisten dengan karakteristik sinyal rPPG di mana sinyal pernapasan cenderung lebih dominan dalam perubahan volume darah di wajah.
4. **Pesan Status:** "Wajah terdeteksi dalam kotak hijau. Silakan diam dan tetap di posisi." Ini mengkonfirmasi keberhasilan deteksi wajah dan memberikan instruksi kepada pengguna untuk menjaga kestabilan posisi.

#### 4.4.2 Bagian Tengah Kiri (Parameter Filter):

1. Ini adalah panel untuk mengatur parameter filter. Pada gambar ini, filter telah diterapkan atau siap diterapkan.
2. **Respirasi Order Filter: 2:** Menunjukkan orde filter yang digunakan untuk sinyal pernapasan. Orde filter mempengaruhi karakteristik respons frekuensi filter.
3. **PPG Low Cut (Hz): 0.8:** Frekuensi potong rendah dari filter untuk sinyal PPG (Photoplethysmography). Sinyal di bawah 0.8 Hz akan dilemahkan. Ini adalah batas bawah yang wajar untuk detak jantung (sekitar 48 BPM).
4. **PPG High Cut (Hz): 2.5:** Frekuensi potong tinggi dari filter untuk sinyal PPG. Sinyal di atas 2.5 Hz akan dilemahkan. Ini adalah batas atas yang wajar untuk detak jantung (sekitar 150 BPM).
5. **PPG Order Filter: 3:** Menunjukkan orde filter yang digunakan untuk sinyal PPG.
6. **Centang: "Wajah stabil di kotak selama 30 detik. Pengukuran detak jantung dan pernapasan dapat dipercaya.":** Ini adalah indikator penting bahwa sistem telah berhasil mendeteksi wajah secara stabil untuk durasi yang cukup lama, sehingga hasil pengukuran detak jantung dan pernapasan dianggap andal.

#### 4.4.3 Bagian Kanan (Grafik Visualisasi Sinyal):

##### 1. Grafik Atas: Sinyal Pernapasan

- **Sumbu Y (Amplitudo):** Menunjukkan kekuatan atau besaran sinyal.
- **Sumbu X (Waktu (detik)):** Menunjukkan durasi waktu pengukuran.
- **Isi Grafik:** Grafik menampilkan gelombang yang berfluktuasi secara periodik, merepresentasikan sinyal pernapasan yang terdeteksi. Bentuk gelombang ini menunjukkan siklus inspirasi dan ekspirasi.

##### 2. Grafik Bawah: Sinyal Detak Jantung (rPPG)

- **Sumbu Y (Amplitudo):** Menunjukkan kekuatan atau besaran sinyal.
- **Sumbu X (Waktu (detik)):** Menunjukkan durasi waktu pengukuran.



- **Isi Grafik:** Grafik ini menampilkan gelombang pulsa yang lebih cepat dan berulang, khas sinyal PPG. Puncak-puncak gelombang ini merepresentasikan detak jantung. Meskipun amplitudonya lebih kecil dari sinyal pernapasan (seperti yang terlihat pada Amplitudo Detak Jantung: 0.003), sinyal ini terdeteksi dengan jelas.

#### 4.4.4 Status Sinyal Setelah 30 Detik:

- **Pernapasan: Bagus - Wah, napasnya dalam dan kuat! Ini sinyal sehat banget, lanjutkan!:** Ini adalah interpretasi kualitatif positif terhadap kualitas sinyal pernapasan.
- **Detak Jantung (PPG): Kurang Bagus - Detak jantungnya kayak malu-malu, coba perbaiki pencahayaan atau posisikan wajah lebih jelas.:** Meskipun detak jantung terdeteksi, program memberikan umpan balik bahwa kualitas sinyal detak jantung kurang optimal. Ini menyarankan perbaikan kondisi akuisisi seperti pencahayaan atau posisi wajah untuk mendapatkan sinyal yang lebih kuat.

#### 4.4.5 Info Sains Singkat:

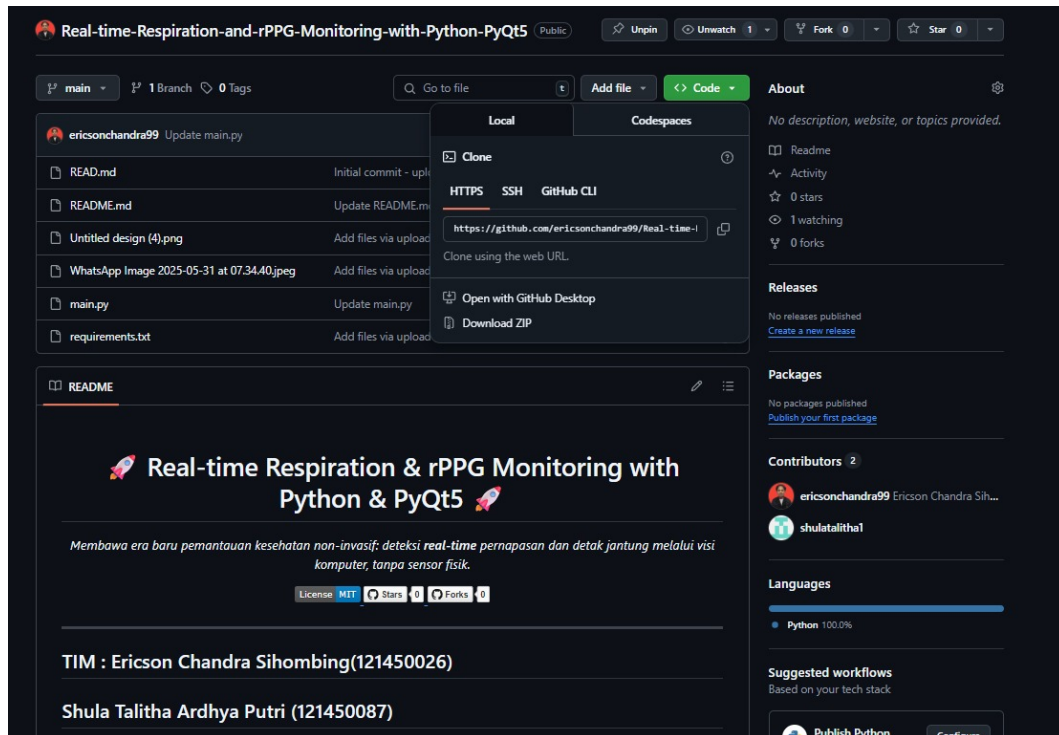
- **Sinyal Pernapasan:** Amplitudo sinyal menandakan intensitas napas. Semakin besar, semakin dalam napas Anda.
- **Sinyal rPPG (Detak Jantung):** Amplitudo menunjukkan kekuatan sinyal detak jantung wajah.
- **Sinyal stabil dan cukup kuat penting untuk hasil akurat.**
- **Pastikan pencahayaan dan posisi wajah baik untuk hasil terbaik.**

## 5 Repositori GitHub dan README.md Proyek

Repositori GitHub digunakan sebagai sarana utama untuk publikasi dan manajemen kode sumber proyek "Real-time Respiration dan rPPG Monitoring with Python dan PyQt5". Penempatannya di platform ini bertujuan untuk mencapai beberapa hal krusial: pertama, memfasilitasi akses global terhadap kode proyek, memungkinkan pihak-pihak berkepentingan untuk mengunduh, mengevaluasi, dan mereplikasi solusi yang telah dikembangkan. Kedua, GitHub berperan sebagai pusat kolaborasi, menyediakan alat kontrol versi yang esensial untuk melacak perubahan, mengelola kontribusi dari tim (dan potensi kontributor eksternal), serta memastikan integritas dan keberlanjutan pengembangan proyek. Lebih lanjut, repositori ini juga berfungsi sebagai representasi profesional atas kapabilitas teknis tim pengembang.

"Monitoring with Python dan PyQt5" di platform kolaborasi global. Melengkapi fungsi repositori, file README.md memegang peranan vital sebagai dokumentasi awal dan panduan komprehensif bagi setiap pengunjung. Tujuan utamanya adalah menyampaikan esensi proyek secara efisien; ini mencakup penjelasan singkat mengenai tujuan aplikasi, urgensi pengembangannya, serta daftar fitur-fitur unggulan yang ditawarkan. Selain itu, README.md menyediakan instruksi detail mengenai prosedur kloning repositori, pembuatan virtual environment, dan langkah-langkah menjalankan aplikasi, yang sangat krusial untuk pengalaman pengguna yang lancar. Inklusi visualisasi (screenshot) juga membantu calon pengguna atau evaluator untuk memahami fungsionalitas dan antarmuka aplikasi secara intuitif, menjadikan README.md sebagai alat komunikasi yang tidak terpisahkan dalam ekosistem proyek sumber terbuka.

Tangkapan layar pada Gambar 2 halaman utama repositori GitHub, berfungsi sebagai bukti visual dan validasi konkret atas keberadaan dan pengelolaan proyek "Real-time Respiration dan rPPG".



Gambar 2: Halaman Repositori GitHub)

## 6 Kesimpulan Proyek

Proyek ini berhasil mengembangkan aplikasi pemantauan kesehatan non-invasif yang mengintegrasikan pengukuran sinyal pernapasan dan detak jantung (rPPG) secara real-time menggunakan webcam dengan pemrosesan sinyal digital yang optimal, dilengkapi antarmuka GUI interaktif berbasis PyQt5. Aplikasi mampu mendeteksi dan menampilkan sinyal biometrik secara akurat dengan filter bandpass yang dapat disesuaikan, memberikan visualisasi grafik dinamis serta informasi kualitas sinyal dan status posisi wajah. Dengan fitur pengaturan parameter dan umpan balik pengguna yang jelas, sistem ini tidak hanya berfungsi sebagai alat edukasi pengolahan sinyal digital, tetapi juga memberikan manfaat praktis dalam pemantauan kesehatan yang mudah diakses dan dapat diandalkan, asalkan pengguna menjaga posisi wajah stabil dan pencahayaan optimal selama pengukuran.

## 7 Referensi dan Daftar Pustaka Proyek

### 7.1 Obrolan AI

[Analisis Visualisasi Aplikasi rPPG](#)

[Kode Real-time Respirasi dan rPPG](#)

[Penyempurnaan GUI Pemrosesan Video](#)

[Metode dan Perhitungan Monitoring Pernapasan dan Detak Jantung Real-Time Menggunakan Webcam](#)

### 7.2 Link Github

[Link Github: Real-time-Respiration-and-rPPG-Monitoring-with-Python-PyQt5](#)