

Evaluating a Preprocessing Pipeline for Fetal Phonocardiography Using FIR Filtering

Article History:

Received
24 February 2025
Revised
26 March 2025
Accepted
26 April 2025

**SAHI RAFAEL DAMARDHI, IRMALIA SURYANI
FARADISA, SOTYOHADI**

Teknik Elektro Institut Teknologi Nasional, Malang, Indonesia
Email: irmalia_suryani_faradisa@lecturer.itn.ac.id

ABSTRAK

Penelitian ini mengevaluasi efektivitas pipeline prapemrosesan yang terdiri dari resampling, normalisasi, dan filtering Finite Impulse Response (FIR) untuk meningkatkan konsistensi sinyal sebelum analisis lanjutan seperti. Resampling dilakukan dengan menyeragamkan laju sampel ke 16 kHz; memastikan resolusi temporal yang seragam. Normalisasi diterapkan untuk menyelaraskan amplitudo antar rekaman, menghasilkan mean 0,0015 dan standar deviasi 0,0462. FIR filtering digunakan untuk mereduksi derau, menghilangkan 77,69% energi sinyal di atas 200 Hz dengan retensi sinyal utama sebesar 29,65%. Evaluasi pipeline menunjukkan Signal-to-Noise Rasio (SNR) -1,88 dB, yang mengindikasikan penurunan daya sinyal, tetapi normalisasi memastikan amplitudo tetap stabil. Hasil ini menunjukkan bahwa kombinasi prapemrosesan ini efektif dalam mengurangi derau, meskipun keseimbangan antara reduksi derau dan preservasi informasi sinyal utama tetap menjadi tantangan.

Kata kunci: *Prapemrosesan Sinyal, FPCG, Normalisasi Min-Max, Finite Impulse Response, Signal to Noise Rasio*

ABSTRACT

This study evaluates the effectiveness of a preprocessing pipeline consisting of resampling, normalization, and Finite Impulse Response (FIR) filtering to improve signal consistency for further signal analyses such as feature extraction and classification. Resampling standardizes the sampling rate to 16 kHz, ensuring uniform temporal resolution. Normalization adjusts amplitude across recordings, yielding a mean of 0,0015 and a standard deviation 0,0462. FIR filtering reduces noise, eliminating 77,69% of signal energy above 200 Hz while retaining 29,65% of the main signal. Pipeline evaluation shows a Signal-to-Noise Ratio (SNR) of -1,88 dB, indicating a significant power reduction, but normalization ensures amplitude stability. These results demonstrate that this preprocessing combination effectively reduces noise, although balancing noise reduction and signal preservation remains challenging.

Keywords: *Preprocessing, Fetal Phonocardiography, Min-max Normalization, Finite Impulse Response, Signal to Noise Ratio*

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

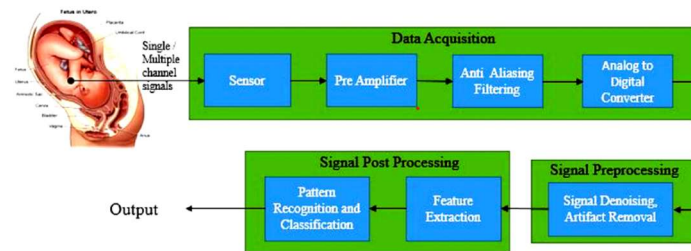


1. PENDAHULUAN

Fetal Phonocardiography (FPCG) adalah teknik pemantauan aktivitas mekanis jantung janin yang non-invasif, pasif, dan ekonomis (**Martinek, dkk, 2020**). FPCG sudah ditemukan sejak abad ke 17, tetapi penelitian tentang FPCG yang melewati proses peer review pada basis data Institute of Electrical and Electronics Engineers (IEEE), Science Direct, dan National Institute of Health (PubMed) masih terbatas, yaitu 35 artikel terpublikasikan pada rentang 2001-2010, dan 33 artikel pada rentang 2011-2016 (**Chetlur, dkk, 2017**). Apabila dibandingkan dengan teknik pemantauan ultrasonik yang memaparkan gelombang ultrasonik secara aktif pada janin, maka FPCG adalah relatif lebih aman, dan dapat dilakukan secara kontinu tanpa penyesuaian letak transducer (**Chetlur, dkk, 2017**). FPCG diimplementasikan dengan menggunakan sensor akustik/stetoskop yang diletakkan pada permukaan perut ibu untuk menangkap sinyal suara detak jantung janin yang kemudian dapat direkam dan disimpan sebagai data audio untuk dapat dianalisis lebih lanjut (**Samieinasab & Sameni, 2015**). Kelemahan mendasar dari teknik FPCG adalah kontaminasi derau seperti gerakan ibu, aktivitas fisiologis ibu, gerakan janin, dan gangguan lingkungan (**Chetlur, dkk, 2017**) (**Faradisa, dkk, 2020**) (**Sbrollini, dkk, 2017**). Derau-derau ini dapat mengaburkan fitur-fitur penting dari sinyal FPCG sehingga sulit untuk dilakukan analisis dan digunakan sebagai set data pengembangan sistem diagnosis terautomasi.

Automasi interpretasi sinyal fonokardiografi memiliki tren yang menjanjikan. Pada kompetisi Physionet 2022 Challenge, tim Listen2YourHeart melatih model *Deep Neural Network* (DNN) secara *Self-Supervised Learning* (SSL), dan model tersebut mampu meraih akurasi sebesar 0,737 dari skala maksimal 1,0 dalam deteksi murmur jantung (**Ballas, dkk, 2022**). Pada penelitian lain sebuah model automasi dengan pendekatan *Artificial Neural Network* (ANN) dan algoritma pelatihan Lavenberg-Marquardt serta *Short Time Fourier Transform* (STFT) mampu mengidentifikasi kelas murmur jantung holodistolik dengan tingkat akurasi 99,99% (**Faradisa, dkk, 2016**). Pendeteksian *arrhythmia* pada detak jantung janin berbasis Ensemble Feature dan Principal Component Analysis (PCA) mampu mencapai akurasi 91,7% (**Faradisa, dkk, 2023**). Terdapat banyak penelitian serupa dengan set data PCG anak-anak dan dewasa yang tersedia secara masif dan terlabel pada platform terbuka Physionet. Namun, untuk sinyal FPCG, set data yang tersedia masih sangat terbatas dan belum terlabel. Pada penelitian ini, set data sinyal FPCG yang dirilis oleh Universitas Shiraz (**Sameni & Samieinasab, n.d.**) dipilih sebab penelitian-penelitian lain terkini pada pengolahan FPCG, seperti (**Faradisa, dkk, 2023**) (**Tomassini, dkk, 2019**) (**Sbrollini, dkk, 2017**) menggunakan set data tersebut. Minimnya penelitian pada prapemrosesan sinyal FPCG menyebabkan analisis lanjutan yang menggunakan teknik klasifikasi atau pendeteksian kondisi patologis pada janin terbatas.

Sebelum sinyal pada set data dapat diolah menggunakan algoritma pembelajaran mesin atau ANN, dibutuhkan tahapan prapemrosesan sinyal. Secara keseluruhan, pengolahan sinyal di tahapan prapemrosesan dapat memberikan sumbangsih pada kerangka pemrosesan sinyal FPCG seperti yang ditampilkan pada Gambar 1.



Gambar 1. Blok Diagram Analisis Sinyal Fonokardiogram Janin (Chetlur, dkk, 2017)

Berdasarkan Gambar 1, tahapan prapemrosesan sinyal memiliki tujuan menghilangkan derau dan artefak sinyal. Sinyal yang telah melewati tahapan prapemrosesan baru dapat digunakan pada pemrosesan lanjutan sinyal seperti ekstraksi fitur, rekognisi pola, dan klasifikasi. Terdapat banyak teknik prapemrosesan pada sinyal, dan pada penelitian ini, akan digunakan teknik prapemrosesan *resampling*, normalisasi min-max, dan implementasi filter *Finite Impulse Response* (FIR). Kombinasi ini merupakan teknik prapemrosesan yang dipilih karena dapat meningkatkan kualitas sinyal tanpa pemrosesan berlebihan. Beberapa metode ekstraksi fitur, seperti *Wavelet Scattering Transform* (WST) akan rentan terhadap redundansi apabila digabungkan dengan teknik denoising (Bruna & Mallat, 2013) (Sbrollini, dkk, 2017), sehingga dibutuhkan teknik prapemrosesan yang lebih sederhana seperti kombinasi yang dipakai pada penelitian ini.

Resampling adalah proses penyesuaian frekuensi sampel sinyal untuk memastikan bahwa keseluruhan sinyal fPCG dari dataset memiliki sampling rate (laju sampel) yang seragam (Vetterli, dkk, 2014). Keseragaman sampling rate berguna dalam mempertahankan konsistensi ekstraksi fitur, khususnya dengan mencegah *aliasing* ketika sinyal diolah. Besar laju sampel ditentukan dengan mengikuti teorema Nyquist-Shanon yang diekspresikan secara matematis pada Persamaan (1):

$$F_s \geq 2F_{\max} \quad (1)$$

F_s adalah frekuensi sampling (sampel per detik) dalam satuan Hz. $F_{\max}$ adalah frekuensi maksimal pada sinyal sampel. Dua kali nilai $F_{\max}$ adalah nilai Nyquist (jumlah minimum nilai laju sampling). Dengan kata lain, laju sampling bernilai setidaknya dua kali nilai tertinggi dari frekuensi sinyal yang disampling agar tidak terjadinya kehilangan informasi pada sinyal. Pada set data FPCG Universitas Shiraz, laju sampling terendah adalah 8 kHz, dan tertinggi 16 kHz. Sesuai Persamaan (1), seluruh sinyal akan disesuaikan laju samplingnya menjadi 16 kHz. Normalisasi min-max adalah penyamaan skala amplitudo antar sinyal. Normalisasi dibutuhkan sebab rekaman FPCG mentah memiliki amplitudo dengan skala yang bervariasi. Hal ini dapat diakibatkan perbedaan intensitas rekaman dan peletakkan posisi sensor. Normalisasi pada penelitian ini mengubah skala amplitudo sinyal menjadi rentang -1 dan +1, sesuai dengan tahapan normalisasi yang dilakukan pada beberapa studi pemrosesan sinyal lanjutan (Ben Hamza & Sjarif, 2025) (He, dkk, 2021) (Zhang, dkk, 2022). Secara matematis, normalisasi dinyatakan pada Persamaan (2):

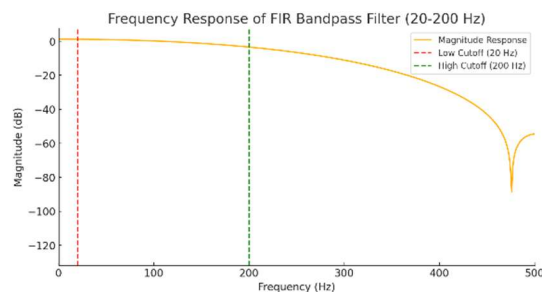
$$X' = a + \frac{(X - X_{\min})(b - a)}{X_{\max} - X_{\min}} \quad (2)$$

X' adalah nilai normalisasi. Variabel a merupakan batas bawah dari rentang yang diinginkan, sedangkan b adalah batas atas. $X_{\min}$ merupakan nilai minimal pada dataset, dan $X_{\max}$ adalah nilai maksimal. Variabel X adalah nilai orisinal. Apabila ingin mendapatkan rentang -1 dan +1,

maka nilai dapat dimasukkan pada Persamaan (2) dan akan didapatkan Persamaan (3) sebagai hasil akhirnya.

$$X' = -1 + \frac{2(X - X_{min})}{X_{max} - X_{min}} \quad (3)$$

FIR adalah filter yang stabil untuk diterapkan pada tahapan prapemrosesan FPCG karena filter ini tidak menerapkan umpan balik pada sistem penyaringannya, proses penyaringan bersifat linear, terkendali, dan dapat menyasar frekuensi pada rentang tertentu tanpa mendistorsi struktur sinyal asli (**Mokeyev, 2013**) (**Johansson & Gustafsson, 2013**). Selain itu, tahap prapemrosesan pada penelitian ini ditujukan agar sinyal dapat diolah lebih lanjut menggunakan teknik analisis lanjutan dengan pendekatan *machine learning* atau ANN, sehingga penerapan filter multi-resolusi kompleks seperti *Discrete Wavelet Transform* (DWT) dan *Wavelet Transform* (WT) bukanlah metode denoising yang tepat untuk diletakkan pada tahapan prapemrosesan. Selain itu WT dan DWT membutuhkan sumberdaya komputasi yang besar karena mekanisme tersebut melibatkan proses pemfilteran yang bersifat berulang dan dapat mendistorsi resolusi sinyal (**Sbrollini, dkk, 2017**) (**Faradisa, dkk, 2020**). FIR, di sisi lain menawarkan filtrasi yang hemat sumberdaya komputasi dan fleksibel apabila hasil akan digunakan dengan pendekatan analisis lanjutan yang berbeda-beda. Suara jantung ibu beserta aktivitas fisiologis tubuh ibu berada di rentang 10-40 Hz, sedangkan suara janin berada pada rentang 20-110 Hz (**Sbrollini, dkk, 2017**) (**Kahankova, dkk, 2023**) (**Farahi, dkk, 2021**). Sinyal dengan nilai frekuensi di atas 200 Hz merupakan derau yang tidak relevan pada analisis FPCG (**Tang, dkk, 2016**). Berdasarkan riset-riset tersebut, ditentukan bahwa FIR pada penelitian ini adalah filter *bandpass* dengan rentang 20 – 200 Hz, sesuai dengan grafik respon frekuensi Gambar 2.



Gambar 2. Grafik Respon Frekuensi Bandpass Filter FIR

Persamaan matematis konvolusi filter FIR pada domain waktu secara umum dapat dinyatakan pada Persamaan (5) di bawah (**Stergiopoulos, 2000**) (**Chen, 2009**):

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n-k] \quad (5)$$

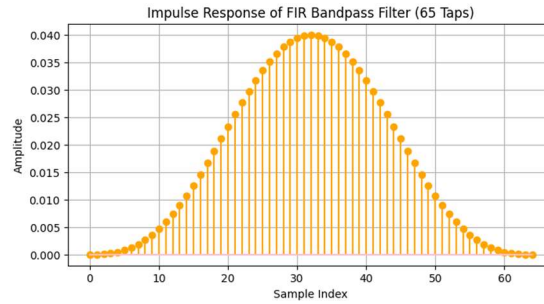
Di mana $y[n]$ adalah sinyal luaran terfilter, $x[n]$ adalah sinyal masukan (FPCG), $h[k]$ adalah respon impuls (koefisien filter) FIR, k adalah indeks *summation* yang merepresentasikan sampel masukan masa lampau, dan N sebagai nilai ordo filter. Dalam fungsi sinc, filter bandpass dengan nilai cut-off frekuensi rendah 20 Hz dan cut-off frekuensi tinggi 200 Hz dapat dinyatakan pada Persamaan (6) di bawah:

$$h[k] = w[k] \left[\frac{\sin \left(2\pi f_H \left(k - \frac{N-1}{2} \right) \right)}{\pi \left(k - \frac{N-1}{2} \right)} - \frac{\sin \left(2\pi f_L \left(k - \frac{N-1}{2} \right) \right)}{\pi \left(k - \frac{N-1}{2} \right)} \right] \quad (6)$$

Suku $y[n]$, $h[k]$, dan variabel N dan k pada Persamaan (6) memiliki keterangan yang sama dengan Persamaan (5). Variabel f_H adalah nilai cutoff frekuensi tinggi, yaitu $f_H = \frac{200}{16000}$; sedangkan f_L adalah nilai cutoff frekuensi rendah, yaitu $f_L = \frac{20}{16000}$. Nilai 16000 didapatkan sesuai Persamaan (1). Suku $w[k]$ adalah fungsi jendela yang dapat dipilih dari 4 pilihan, yaitu Hann, Hanning, Blackman, dan Kaiser. Di penelitian ini window Hann dipilih karena memiliki tansisi ketajaman yang paling halus dan atenuasi dengan nilai negatif terkecil (-31 dB). Window Hann dirumuskan pada persamaan (7):

$$w[k] = 0.54 - 0.46 \cos \left(\frac{2\pi k}{N-1} \right) \quad (7)$$

Respon impuls yang diharapkan dengan desain filter di atas adalah sesuai dengan Gambar 3 di bawah:



Gambar 3. Respon Impuls Ideal FIR Bandpass Filter (20-200 Hz)

Desain bandpass di atas menggunakan filter tap sebesar 65, sehingga titik tengahnya (di sekitaran nilai 30-40) berkorespondensi dengan titik puncak filter. Saat sample index bernilai 20, amplitudo koefisien memiliki nilai yang berkorelasi dengan hasil filtering. Dapat diamati bahwa sample index 20 memiliki amplitudo koefisien yang lebih kecil dibandingkan index 30; ini berarti kontribusi index 20 lebih rendah dibandingkan ketika index bernilai 30.

Efisiensi tahap prapemrosesan pada penelitian ini akan dievaluasi berdasarkan metrik nilai Signal to Noise Ratio (SNR), Reduksi Energi pada Frekuensi ≥ 200 Hz, Retensi Energi pada rentang 20-200 Hz, dan Statistik Amplitudo (*Mean* dan Deviasi Standar Amplitudo) (**Vetterli, dkk, 2014**). SNR mencerminkan peningkatan kualitas sinyal berdasarkan pengurangan derau. Nilai SNR didapat berdasarkan rasio P_{signal} , merepresentasikan intensitas sinyal yang diinginkan, terhadap P_{noise} , atau intensitas derau. Rasio ini dapat dinyatakan secara matematis, sesuai dengan Persamaan (8).

$$SNR = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}} \quad (8)$$

Metrik Reduksi Energi mengevaluasi kemampuan tahap prapemrosesan dalam mengurangi derau berfrekuensi tinggi. Reduksi energi dinyatakan dalam %, sesuai dengan perumusan

Persamaan (9) berikut ini, di mana E_{before} adalah energi sinyal sebelum prapemrosesan, dan E_{after} adalah energi sinyal sesudah prapemrosesan.

$$\%Reduksi\ Energi = \frac{E_{before} - E_{after}}{E_{before}} \times 100 \quad (9)$$

Retensi energi merepresentasikan informasi pada FPCG yang dapat dipertahankan setelah melalui tahap prapemrosesan. Secara matematis, retensi energi dinyatakan pada Persamaan (10). $E_{filtered}$ adalah energi sinyal sesudah terfilter, dan $E_{original}$ merupakan energi sinyal orisinal.

$$\%Retensi\ Energi = \frac{E_{filtered}}{E_{original}} \times 100 \quad (10)$$

Metrik *Mean* (μ) amplitudo merepresentasikan rata-rata level amplitudo sinyal, yang secara matematis dihitung dengan Persamaan (11), di mana N adalah jumlah total sampel, x_i adalah nilai individu sinyal. Nilai μ ideal adalah mendekati 1.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (11)$$

Metrik deviasi standar amplitudo (σ) menyatakan variasi amplitudo terhadap nilai μ . Nilai σ tinggi menunjukkan fluktuasi tinggi pada amplitudo, sedangkan nilai mendekati nol menunjukkan keseragaman sinyal (tidak terdapat banyak fluktuasi). Persamaan (12) merumuskan bagaimana σ didapatkan berdasarkan hubungan di antara variabel N , x_i , dan μ .

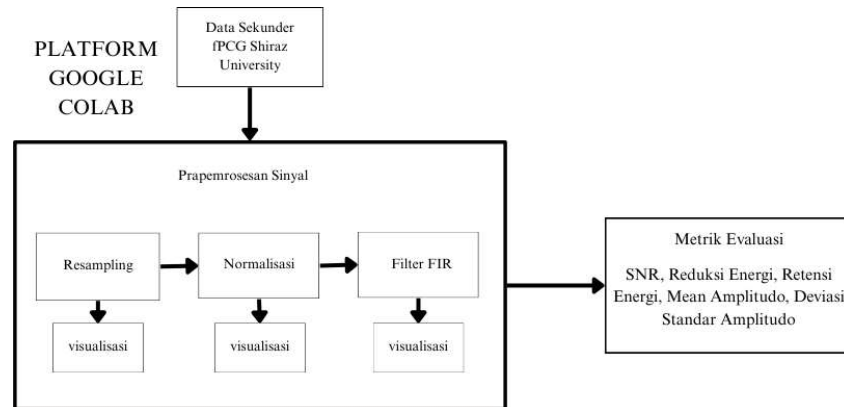
$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (12)$$

Implementasi tahapan prapemrosesan di atas yang melingkupi *resampling*, normalisasi, dan filtering menggunakan FIR dilakukan pada platform Google Collab, sehingga pengembangan program dapat bersifat terbuka (*open-source*), dan mudah untuk dikembangkan lebih lanjut. Set data yang sudah diolah akan tersedia secara publik pada platform GitHub.

2. METODE

2.1 Perancangan Sistem Prapemrosesan Sinyal

Sistem prapemrosesan dirancang agar dapat diimplementasikan pada platform Google Collab. Gambar 4 di bawah adalah diagram blok sistem prapemrosesan sinyal di penelitian ini:



Gambar 4. Blok Diagram Sistem Prapemrosesan Sinyal

Sesuai dengan Gambar 4, set Data FPCG Shiraz University diunggah pada penyimpanan awan Google Drive sehingga proses komputasi pada Google Collab tidak memerlukan pengunggahan ulang di setiap sesinya ketika terjadi pengulangan. Terdapat 120 sinyal dengan format .wav dengan format penamaan fxx, di mana f mengindikasikan rekaman janin, dan xx adalah nomor subjek. Di setiap tahap prapemrosesan, akan ditampilkan visualisasi dari sinyal agar observasi figur sinyal dapat dilakukan. Metrik Evaluasi menampilkan evaluasi kualitas prapemrosesan berdasarkan 5 parameter, yaitu SNR, Reduksi energi, Retensi Energi, Mean Amplitudo, dan Deviasi Standar Amplitudo.

2.2 Resampling

Fungsi *resampling* dalam pemrograman ditulis sesuai dengan pseudocode Gambar 5 di bawah:

```

DEFINE FUNCTION resample_audio (input_path, output_path, target_sampling_rate = 16000 Hz):
  TRY:
    LOAD audio file from input_path and get original_sampling_rate
    PRINT original sampling rate of the audio file
    IF original_sampling_rate ≠ 16000 Hz:
      RESAMPLE audio to 16000 Hz
      PRINT message confirming successful resampling to 16000 Hz
      SAVE resampled audio to output_path
      SAVE original audio as numpy file for visualization
      SAVE resampled audio as numpy file for visualization
      SAVE sampling rates (original and 16000 Hz) as numpy file
      PRINT message confirming saved resampled audio
    ELSE:
      SAVE audio directly without resampling
      PRINT message confirming file already at 16000 Hz; no resampling required
  CATCH any errors:
    PRINT error message describing the issue with input_path
  END FUNCTION

```

Gambar 5. Pseudocode Resampling

Program *resampling* pada Gambar 5 memiliki 4 fungsi utama, yaitu memeriksa apakah file audio perlu *di-resampling*, melakukan *resampling* ke target laju sampling 16 kHz, menyimpan file hasil *resampling* dan data tambahan (*numpy arrays*) untuk analisis lebih lanjut (berupa visualisasi), dan menghindari *resampling* yang tak perlu dengan menyalin langsung file yang

sudah memiliki laju sampling 16 kHz. Setelah *resampling*, visualisasi figur sinyal dijalankan dengan pseudocode di Gambar 6:

```

DEFINE FUNCTION plot_signals (original_signal, resampled_signal, original_sr, target_sr, filename):
  INITIALIZE new figure window for plotting
  PLOT original_signal in subplot:
    Set plot color and transparency
    Set title indicating "Original Signal", filename, and original sampling rate
    Label x-axis ("Samples") and y-axis ("Amplitude")
  PLOT resampled_signal in subplot:
    Set plot color and transparency
    Set title indicating "Resampled Signal", filename, and target sampling rate (e.g., 16000 Hz)
    Label x-axis ("Samples") and y-axis ("Amplitude")
  ADJUST layout to avoid overlap
  DISPLAY plots
END FUNCTION
SET processed_directory (directory containing resampled .npz files)
FIND all "_original.npz" files in processed_directory
IF no "_original.npz" files found:
  PRINT warning message to run resampling first
ELSE:
  PRINT number of resampled files found for visualization
  FOR each "_original.npz" file:
    EXTRACT base filename by removing "_original.npz" suffix
    LOAD original_signal from "{base_name}_original.npz"
    LOAD resampled_signal from "{base_name}_resampled.npz"
    LOAD original and target sampling rates from "{base_name}_sr.npz"
    CALL plot_signals(original_signal, resampled_signal, original_sr, target_sr, base_name)
  END FOR
  PRINT "Visualization completed!"
ENDIF

```

Gambar 6. Pseudocode Visualisasi Hasil *Resampling*

Program visualisasi (Gambar 6) menggambarkan sinyal FPCG dalam bentuk grafik. Fungsi `plot_signals ()` membuat dua subplot, di mana sinyal asli ditampilkan dalam warna biru, dan sinyal yang telah di *resampling* akan ditampilkan berwarna hijau, dilengkapi dengan keterangan laju sampling pada masing-masing grafik. Program kemudian mencari file NumPy (.npz) yang telah disimpan dari proses *resampling* untuk dimuat, lalu divisualisasikan. Apabila file ditemukan, sinyal asli dan sinyal hasil *resampling* setiap rekaman FPCG akan ditampilkan satu per satu.

2.3 Normalisasi

Fungsi normalisasi pada program, yang ditulis berdasarkan Persamaan (4) adalah sebagai berikut:

```
DEFINE FUNCTION min_max_normalize(audio_signal, range_min = -1, range_max = 1):  
  FIND minimum value of audio_signal (min_val)  
  FIND maximum value of audio_signal (max_val)  
  COMPUTE normalized_signal as:  
    normalized_signal = (audio_signal - min_val) / (max_val - min_val)  
    normalized_signal = normalized_signal * (range_max - range_min) + range_min  
  RETURN normalized_signal  
END FUNCTION
```

Gambar 7. Pseudocode Normalisasi Min-Max Rentang -1 dan +1

Pada Gambar 7, program melakukan normalisasi min-max dengan skala -1 dan +1 pada sinyal. Program mengunduh sinyal FPCG terlebih dahulu, lalu menelusuri nilai minimal dan maksimal pada sinyal. Setelah itu, sinyal diubah ke skala 0 dan 1 terlebih dahulu sebelum kemudian diubah ke skala -1 dan +1, sesuai parameter $range_min=-1$ dan $range_max=1$. Visualisasi sinyal yang sudah melewati proses normalisasi dijalankan dengan program pada pseudocode Gambar 8 di bawah ini:

```
DEFINE FUNCTION compute_impact_ratio(file, input_dir, output_dir):  
  LOAD original and normalized signals  
  RETURN (max(abs(normalized)) / max(abs(original))) IF max(original)  $\neq$  0 ELSE 0  
END FUNCTION  
DEFINE FUNCTION visualize_top_affected_signals(file_list, input_dir, output_dir, top_n):  
  COMPUTE and SORT files by impact_ratio (descending)  
  FOR each of top_n files:  
    LOAD original and normalized signals (resample if needed)  
    PLOT both signals amplitude vs. time  
    DISPLAY plots  
  END FOR  
END FUNCTION  
CALL visualize_top_affected_signals(df_comparison, input_directory, output_directory, top_n=5)  
PRINT "Visualization completed!"
```

Gambar 8. Pseudocode Visualisasi Sinyal-Sinyal Paling Terdampak Oleh Normalisasi

Program pada Gambar 8 memvisualisasikan sinyal yang paling terdampak oleh normalisasi dengan membandingkan sinyal asli dan sinyal hasil normalisasi. Fungsi pada program ini menerima DataFrame yang berisi nama file dan rasio dampak normalisasi (*Impact Ratio*), lalu memilih sinyal sejumlah *top-n* yang paling terpengaruh.

2.4 Filter FIR

Program FIR sesuai Persamaan (5) dan (6) diimplementasikan pada program Gambar 9 di bawah:

```
SET sampling_rate = 16000 Hz  
SET low_cutoff = 20 Hz  
SET high_cutoff = 200 Hz  
SET filter_order = 64 (adjustable)  
SET window_type = "hann"  
CALCULATE nyquist_rate = sampling_rate / 2  
COMPUTE FIR filter coefficients:  
  - filter_order taps
```

```

- normalized cutoff frequencies = [low_cutoff / nyquist_rate, high_cutoff / nyquist_rate]
- window = window_type
- pass_zero = False (bandpass filter)
DEFINE FUNCTION apply_fir_filter (audio_signal, fir_coefficients):
    RETURN filtered audio_signal using FIR filter
END FUNCTION
FOR EACH audio_file IN input_directory:
    LOAD audio at sampling_rate
    APPLY FIR filter
    SAVE filtered audio to output_directory
END FOR
PRINT "Filtering complete. Filtered audio saved to output directory."

```

Gambar 9. Pseudocode Filter FIR

FIR bandpass pada program Gambar 9 dirancang untuk menjaga komponen detak jantung janin yang berada di rentang 20-200 Hz, sekaligus meredam gangguan yang berada di luar rentang tersebut. Dengan laju sampling 16 kHz, dan ordo filter 64, FIR menghasilkan 65 taps, sesuai grafik respons impuls Gambar 3. Fungsi jendela Hann dipilih sesuai pemahaman Persamaan (7). Gambar 10 di bawah adalah program visualisasi sinyal FPCG sejumlah n yang paling terdampak oleh filter FIR:

```

DEFINE FUNCTION calculate_energy(signal):
    RETURN sum(signal2)
END FUNCTION
DEFINE FUNCTION get_most_affected_signals(input_dir, filtered_dir, top_n):
    FOR each audio file in input_dir:
        LOAD original and filtered signals
        COMPUTE energy_original and energy_filtered
        COMPUTE energy_difference = abs(energy_original - energy_filtered)
        STORE filename and energy_difference
    ENDFOR
    SORT files by energy_difference descending
    RETURN top_n most affected files
END FUNCTION
DEFINE FUNCTION visualize_most_affected(input_dir, filtered_dir, top_n):
    GET top_n files using get_most_affected_signals
    FOR each file:
        LOAD original and filtered signals
        PLOT spectrogram of original signal
        PLOT spectrogram of filtered signal
        DISPLAY plots
    END FOR
END FUNCTION
CALL visualize_most_affected(input_directory, filtered_directory, top_n=5)

```

Gambar 10. Program Visualisasi Sinyal setelah Filter FIR

Pada Gambar 10, sinyal paling terdampak didapatkan dengan membandingkan spektrogram sinyal sebelum dan sesudah implementasi filter FIR. Fungsi `get_most_affected_signals ()`

dipakai untuk identifikasi sinyal dengan perubahan paling signifikan, kemudian setiap sinyal diunduh dan divisualisasikan dalam spektrogram dan diurutkan berdasarkan energi sinyal.

2.5 Metrik Evaluasi

Metrik evaluasi terdiri dari parameter SNR, Reduksi & Retensi Energi, dan mean serta deviasi standar amplitudo. Parameter-parameter tersebut diimplementasikan berdasarkan program pada pseudocode Gambar 11 di bawah:

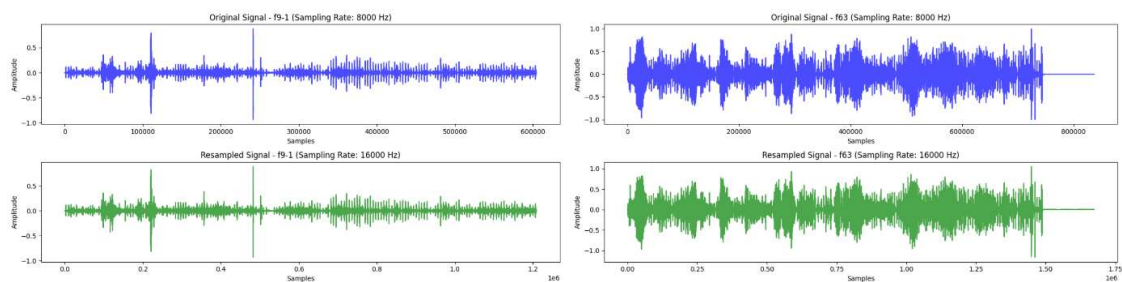
```
DEFINE FUNCTION compute_snr(original, filtered):  
    RETURN 10*log10(signal_power/noise_power)  
END FUNCTION  
DEFINE FUNCTION compute_spectral_energy(signal, sr, fmin, fmax):  
    COMPUTE energy within frequency range [fmin, fmax] from STFT  
    RETURN energy  
END FUNCTION  
FOR EACH audio file IN normalized_dir:  
    IF filtered version exists:  
        LOAD normalized and filtered signals  
        COMPUTE:  
            - SNR improvement  
            - Energy reduction (>200Hz)  
            - Energy retention (20–200Hz)  
            - Mean and Std of amplitude  
        STORE computed values  
END FOR  
COMPUTE averages of all stored values  
PRINT average results:  
    - SNR improvement  
    - High-frequency energy reduction  
    - Energy retention in target band (20-200Hz)  
    - Mean and Std amplitude after normalization
```

Gambar 11. Program Metrik Evaluasi Prapemrosesan

3. HASIL DAN PEMBAHASAN

3.1 Hasil *Resampling*

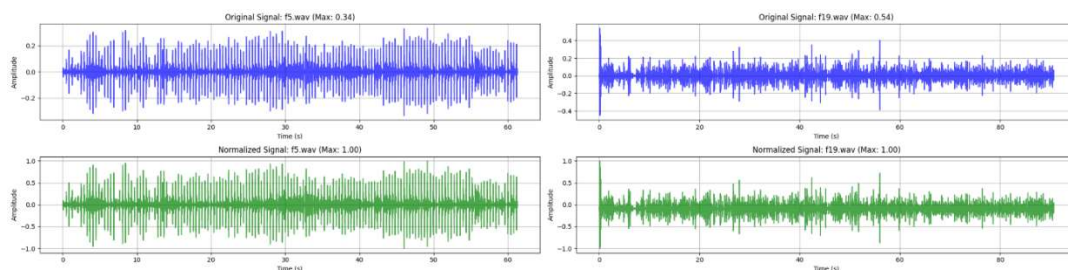
Dari seluruh 120 sinyal FPCG pada dataset, Sinyal f9-1 (menandakan subjek adalah salah satu dari dua janin kembar) dan f63 adalah dua sinyal dengan laju sampling orisinal 8 kHz dan disesuaikan menjadi 16 kHz. Pada Gambar 12, dapat dilihat pada sumbu x di kedua grafik berwarna biru, samples yang awalnya berada pada kisaran nilai 600000 dan 800000 sampel berubah menjadi skala 1200000 dan 1750000 sampel. Dua grafik ini mewakili keseluruhan proses *resampling*, dan dapat diamati bahwa penyesuaian laju sampling telah berhasil.



Gambar 12. Figur Sinyal Sebelum dan Sesudah *Resampling*

3.2. Hasil Normalisasi

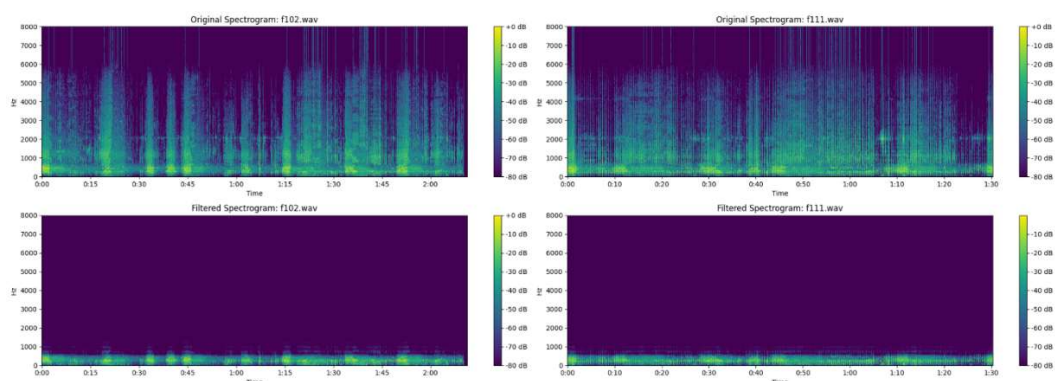
Pada proses normalisasi, sinyal f5 dan f19 adalah dua sinyal paling terdampak oleh normalisasi. Hal ini dapat diamati pada Gambar 13, di mana sinyal f5 dan f19 memiliki amplitudo maksimal orisinal sebesar 0,34 dan 0,54 secara berurutan, dan setelah proses normalisasi, kedua amplitudo secara seragam disesuaikan amplitudonya sehingga berada di rentang +1 dan -1. Visualisasi ini mewakili normalisasi di seluruh sinyal FPCG pada set data.



Gambar 13. Figur Sinyal Sebelum dan Sesudah Normalisasi

3.3 Hasil Filter FIR

Filter FIR berhasil mereduksi komponen sinyal berfrekuensi tinggi, dan Gambar 14 menunjukkan dua sinyal, yaitu f102 dan f111, yang paling terdampak oleh proses filtrasi. Pada kedua sinyal, sebaran energi berfrekuensi tinggi dengan nilai lebih dari 1000 Hz dan kurang dari sama dengan 8000 Hz, digambarkan dengan spektrum berwarna kuning, tersebar di sepanjang waktu sinyal. Setelah filtrasi, tidak terdapat frekuensi berenergi +0 dB (tidak teratenuasi) yang melebihi skala 1000 Hz.

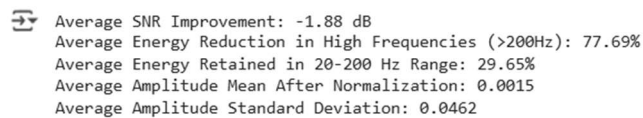


Gambar 14 Spektrogram Sinyal Sebelum dan Sesudah Filter FIR

3.4 Metrik Evaluasi

Prapemrosesan pada penelitian ini menghasilkan sinyal FPCG dengan nilai rata-rata perbaikan SNR sebesar -1,88 dB, Reduksi Energi pada frekuensi tinggi lebih dari 200 Hz sebesar 77,69%,

Retensi energi di rentang 20-200 Hz sebesar 29,65%, μ bernilai 0,0015, dan σ bernilai 0,042. Gambar 15 di bawah menunjukkan hasil tangkap layar dari hasil perhitungan Persamaan (8-12) yang diimplementasikan melalui program (tertulis pada Gambar 11):



```

Average SNR Improvement: -1.88 dB
Average Energy Reduction in High Frequencies (>200Hz): 77.69%
Average Energy Retained in 20-200 Hz Range: 29.65%
Average Amplitude Mean After Normalization: 0.0015
Average Amplitude Standard Deviation: 0.0462

```

Gambar 15. Luaran Program Metrik Evaluasi

Reduksi energi menunjukkan keberhasilan tahap prapemrosesan dalam menyaring 77,69% derau dari keseluruhan sinyal. Namun, nilai SNR yang negatif, yaitu -1,88 dB, dan retensi energi yang jauh dari 100%, yaitu 29,65% mengindikasikan adanya komponen sinyal yang teratenuasi bersamaan dengan derau. Nilai mean (μ) dan standar deviasi amplitudo (σ) yang mendekati nol menandakan normalisasi berhasil secara efektif menyeragamkan amplitudo pada seluruh sinyal. Prapemrosesan ini adalah salah satu tahap dari keseluruhan *pipeline* yang akan melibatkan analisis lanjutan seperti ekstraksi fitur dan klasifikasi, sehingga nilai SNR dan retensi energi bukanlah nilai akhir yang dapat mewakili performa *pipeline* prapemrosesan saat digabungkan dengan sistem analisis yang utuh.

4. KESIMPULAN

Metrik evaluasi menunjukkan penerapan filter FIR dapat mencapai reduksi energi hingga 77,69% pada energi dengan frekuensi di atas 200 Hz. Metode ini efektif dalam menekan komponen noise yang tidak diinginkan. Energi yang dapat dipertahankan setelah prapemrosesan adalah sebesar 29,65%, menunjukkan adanya kemungkinan informasi detak jantung janin yang tereduksi. SNR bernilai -1,88 dB, mengindikasikan sinyal setelah proses filter memiliki daya yang lebih rendah secara signifikan ketika dibandingkan dengan daya derau yang tersisa. Di sisi lain, normalisasi dengan nilai *mean* dan deviasi standar sebesar 0,0015 dan 0,0462 secara berurutan, memastikan antar rekaman memiliki konsistensi keseragaman amplitudo. Metrik-metrik ini bukanlah indikator akhir dari efektivitas prapemrosesan, karena tujuan utama dari pipeline ini adalah mempersiapkan sinyal FPCG agar dapat diolah pada proses analisis lanjutan, seperti ekstraksi fitur dan klasifikasi.

DAFTAR RUJUKAN

- Ballas, Aristotelis, Papapanagiotou, V., Delopoulos, A., & Diou, C. (2022). *Listen2YourHeart: A Self-Supervised Approach for Detecting Murmur in Heart-Beat Sounds*. 2022 Computing in Cardiology Conference. <https://doi.org/10.22489/CinC.2022.298>
- Ben Hamza, M. F. A., & Sjarif, N. N. A. (2025). Comparative Analysis of Feature Selection Based on Metaheuristic Methods for Human Heart Sounds Classification Using PCG Signal. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 16(1). <https://dx.doi.org/10.14569/IJACSA.2025.0160175>

- Bruna, J., & Mallat, S. (2013). Invariant Scattering Convolution Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1872–1886. <https://doi.org/10.1109/TPAMI.2012.230>
- Chen, W.-K. (Ed.). (2009). *Passive, active, and digital filters* (3. ed). CRC Press.
- Chetlur Adithya, P., Sankar, R., Moreno, W. A., & Hart, S. (2017). Trends in fetal monitoring through phonocardiography: Challenges and future directions. *Biomedical Signal Processing and Control*, 33, 289–305. <https://doi.org/10.1016/j.bspc.2016.11.007>
- Faradisa, I. S., Anggriawan, D. O., Sardjono, T. A., & Purnomo, M. H. (2016). Identification of phonocardiogram signal based on STFT and Marquart Lavenberg Backpropagation. *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 25–30. <https://doi.org/10.1109/ISITIA.2016.7828628>
- Faradisa, I. S., Putra, O. V., Sardjono, T. A., & Purnomo, M. H. (2023). Arrhythmia Feotus Heartbeat Detection Using Optimized Neural Network Based on Phonocardiograph Ensemble Feature and Principal Component Analysis. *International Journal of Intelligent Engineering and Systems*, 16(1). <https://doi.org/10.22266/ijies2023.0228.48>
- Farahi, M., Casals, A., Sarrafzadeh, O., Zamani, Y., Ahmadi, H., Behbood, N., & Habibian, H. (2021). *Beat-to-Beat Fetal Heart Rate Analysis Using Portable Medical Device and Wavelet Transformation Technique*. <https://doi.org/10.48550/ARXIV.2103.01014>
- He, Y., Li, W., Zhang, W., Zhang, S., Pi, X., & Liu, H. (2021). Research on Segmentation and Classification of Heart Sound Signals Based on Deep Learning. *Applied Sciences*, 11(2), 651. <https://doi.org/10.3390/app11020651>
- Johansson, H., & Gustafsson, O. (2013). Two-Rate Based Structures for Computationally Efficient Wide- Band FIR Systems. In F. P. Garca Mrquez (Ed.), *Digital Filters and Signal Processing*. InTech. <https://doi.org/10.5772/52198>
- Kahankova, R., Mikolasova, M., Jaros, R., Barnova, K., Ladrova, M., & Martinek, R. (2023). A Review of Recent Advances and Future Developments in Fetal Phonocardiography. *IEEE Reviews in Biomedical Engineering*, 16, 653–671. <https://doi.org/10.1109/RBME.2022.3179633>
- Martinek, R., Barnova, K., Jaros, R., Kahankova, R., Kupka, T., Jezewski, M., Czabanski, R., Matonia, A., Jezewski, J., & Horoba, K. (2020). Passive Fetal Monitoring by Advanced Signal Processing Methods in Fetal Phonocardiography. *IEEE Access*, 8, 221942–221962. <https://doi.org/10.1109/ACCESS.2020.3043496>

- Mokeyev, A. (2013). Direct Methods for Frequency Filter Performance Analysis. In F. P. Garca Mrquez (Ed.), *Digital Filters and Signal Processing*. InTech. <https://doi.org/10.5772/52192>
- Sameni, R., & Samieinasab, M. (n.d.). *Shiraz University Fetal Heart Sounds Database* (Version 1.0.1) [Dataset]. PhysioNet. <https://doi.org/10.13026/42EG-8E59>
- Samieinasab, M., & Sameni, R. (2015). Fetal phonocardiogram extraction using single channel blind source separation. *2015 23rd Iranian Conference on Electrical Engineering*, 78–83. <https://doi.org/10.1109/IranianCEE.2015.7146186>
- Sbrollini, A., Strazza, A., Caragiuli, M., Mozzoni, C., Tomassini, S., Agostinelli, A., Morettini, M., Fioretti, S., Di Nardo, F., & Burattini, L. (2017, September 14). *Fetal Phonocardiogram Denoising by Wavelet Transformation: Robustness to Noise*. 2017 Computing in Cardiology Conference. <https://doi.org/10.22489/CinC.2017.331-075>
- Stergiopoulos, S. (Ed.). (2000). *Advanced signal processing handbook: Theory and implementation for radar, sonar, and medical imaging real time systems*. CRC Press.
- Suryani Faradisa, I., Ananda, A., Arief Sardjono, T., & Hery Purnomo, M. (2020). Denoising of Fetal Phonocardiogram Signal by Wavelet Transformation. *E3S Web of Conferences*, 188, 00013. <https://doi.org/10.1051/e3sconf/202018800013>
- Tang, H., Li, T., Qiu, T., & Park, Y. (2016). Fetal Heart Rate Monitoring from Phonocardiograph Signal Using Repetition Frequency of Heart Sounds. *Journal of Electrical and Computer Engineering*, 2016, 1–6. <https://doi.org/10.1155/2016/2404267>
- Tomassini, S., Strazza, A., Sbrollini, A., Marcantoni, I., Morettini, M., Fioretti, S., Burattini, L., 1 Cardiovascular Bioengineering Lab, Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy, & 2 Laboratorio di Bioingegneria, Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy. (2019). Wavelet filtering of fetal phonocardiography: A comparative analysis. *Mathematical Biosciences and Engineering*, 16(5), 6034–6046. <https://doi.org/10.3934/mbe.2019302>
- Vetterli, M., Kovačević, J., & Goyal, V. K. (2014). *Foundations of signal processing*. Cambridge University Press.
- Zhang, L., Lim, C. P., Yu, Y., & Jiang, M. (2022). Sound classification using evolving ensemble models and Particle Swarm Optimization. *Applied Soft Computing*, 116, 108322. <https://doi.org/10.1016/j.asoc.2021.108322>