

APLIKASI FILTERING AUDIO DENGAN PYTHON

RIZKY CAHYONO PUTRA

442023611012

Informatics Engineering, science and technology,
University of Darussalam Gontor
rizkycahyonoputra80@student.cs.unida.gontor.ac.id

M. IRFANSYAH

442023611004

Informatics Engineering, science
and technology, University of
Darussalam Gontor
mirfansyah26@student.cs.unida.gontor.ac.id

SYAIFAN NOER IWAWAN

442023611008

Informatics Engineering, science and
technology, University of Darussalam
Gontor
syaifannoeriwawan78@student.cs.unida.gontor.ac.id

MUHAMMAD GALANG FACHREZY

442023611011

Informatics Engineering, science and technology,
University of Darussalam Gontor
muhammadgalangfachrezy22@student.cs.unida.gontor.ac.id

ACHMAD FATICH AL - FAHMI

422021611002

Informatics Engineering, science and
technology, University of Darussalam
Gontor
achmadfatichalfahmi@student.cs.unida.gontor.ac.id

***Abstrak**—Laporan ini membahas implementasi aplikasi Filtering Sinyal Audio yang bertujuan untuk mengurangi noise atau frekuensi yang tidak diinginkan pada rekaman audio. Aplikasi ini dibangun menggunakan bahasa Python dengan antarmuka grafis berbasis Tkinter, serta mengintegrasikan konsep dasar Pengolahan Sinyal Digital (PSD) seperti analisis domain waktu, analisis domain frekuensi, dan filtering digital. Pengguna dapat mengunggah file audio dalam format WAV, memilih jenis filter (Low-Pass, High-Pass, Band-Pass), serta mengatur frekuensi cut-off secara manual. Hasil filtering divisualisasikan melalui grafik domain waktu dan frekuensi, serta dapat didengar langsung melalui pemutar audio. Berdasarkan pengujian, aplikasi mampu menampilkan perbedaan signifikan antara sinyal asli dan sinyal yang sudah difilter, baik secara visual maupun auditori.*

***Kata Kunci:** Filtering Audio, Pengolahan Sinyal Digital, Low-Pass Filter, High-Pass Filter, Band-Pass Filter, Python..*

I. PENDAHULUAN

Pengolahan Sinyal Digital (PSD) merupakan cabang ilmu yang berfokus pada pemrosesan sinyal menggunakan teknik komputasi digital. Salah satu bentuk sinyal yang paling sering ditemui adalah sinyal audio, yang digunakan dalam berbagai aplikasi seperti telekomunikasi, multimedia, pengenalan suara, dan pemrosesan musik. Dalam kenyataannya, sinyal audio sering kali terkontaminasi oleh noise, gangguan, atau komponen frekuensi yang tidak diinginkan, sehingga kualitas audio menjadi berkurang.

Untuk mengatasi permasalahan tersebut, salah satu teknik yang umum digunakan adalah **filtering**, yaitu proses untuk mempertahankan komponen frekuensi yang diinginkan dan menghilangkan komponen yang tidak diinginkan. Filtering dapat dilakukan dengan berbagai metode, baik secara analog maupun digital. Dalam konteks ini, filtering digital lebih fleksibel dan mudah diimplementasikan pada sistem berbasis komputer.

II. PENJELASAN KODE DASAR

A. Antarmuka Pengguna (GUI)

Antarmuka aplikasi dibangun menggunakan **Tkinter**. GUI mencakup tombol untuk mengunggah file audio, memilih jenis filter, mengatur frekuensi cut-off, menampilkan hasil visualisasi sinyal, dan memutar audio sebelum serta sesudah filtering.

B. Proses Filtering

Filtering dilakukan dengan menggunakan **Butterworth Filter** dari pustaka SciPy. Tiga jenis filter yang diimplementasikan adalah:

- **Low-Pass Filter:** Menghilangkan frekuensi di atas nilai cut-off.
- **High-Pass Filter:** Menghilangkan frekuensi di bawah nilai cut-off.
- **Band-Pass Filter:** Mempertahankan frekuensi dalam rentang tertentu dan meredam lainnya.

C. Analisis Domain Frekuensi

Untuk menampilkan spektrum frekuensi sinyal, aplikasi menggunakan **Fast Fourier Transform (FFT)**. Hasil FFT divisualisasikan menggunakan Matplotlib sehingga pengguna dapat melihat perbedaan sinyal sebelum dan sesudah filtering.

D. Pemutaran Audio

Aplikasi menggunakan **Pydub** dan **FFmpeg** untuk memutar file audio asli maupun hasil filtering.

III. MODIFIKASI YANG DILAKUKAN DAN ALASAN

Beberapa pengembangan dan penyesuaian dilakukan pada kode dasar agar aplikasi lebih interaktif, di antaranya:

1. **Penambahan Berbagai Jenis Filter:** Menyediakan Low-Pass, High-Pass, dan Band-Pass untuk memberikan fleksibilitas kepada pengguna.
2. **Visualisasi Ganda (Domain Waktu & Frekuensi):** Agar pengguna dapat menganalisis perbedaan sinyal secara kuantitatif dan kualitatif.
3. **Fitur Pemutaran Audio:** Untuk memberikan umpan balik langsung terhadap hasil filtering.
4. **Pengaturan Frekuensi Cut-Off:** Pengguna dapat menentukan nilai cut-off secara manual agar hasil filtering sesuai kebutuhan.

IV. HASIL EKSPERIMEN DAN ANALISIS

A. Tampilan Aplikasi

Aplikasi berhasil menampilkan GUI sederhana yang memudahkan pengguna dalam mengunggah file audio, memilih filter, dan mengontrol parameter filtering.

B. Analisis Domain Waktu dan Frekuensi

- **Sinyal Asli:** Menampilkan bentuk gelombang dan spektrum dengan noise dominan.
- **Low-Pass Filter:** Frekuensi tinggi berkurang, noise hilang signifikan.
- **High-Pass Filter:** Frekuensi rendah berkurang, suara menjadi lebih tipis.
- **Band-Pass Filter:** Hanya rentang tertentu yang dipertahankan, cocok untuk pemrosesan vokal.

C. Evaluasi Performa

Filter bekerja sesuai teori. Low-pass efektif untuk mengurangi noise high-frequency, sedangkan high-pass untuk mengurangi hum. Band-pass digunakan untuk mengekstrak komponen suara spesifik.

V. REFLEKSI PRIBADI

Dalam mengerjakan proyek ini, saya memperoleh pemahaman yang lebih mendalam mengenai konsep dasar pengolahan sinyal digital, khususnya filtering audio. Tantangan terbesar adalah memahami desain filter digital agar sesuai dengan karakteristik sinyal. Penggunaan GUI

berbasis Tkinter juga memberikan pengalaman baru dalam mengembangkan aplikasi interaktif. Proyek ini membuktikan bahwa Python sangat fleksibel untuk pengembangan aplikasi pemrosesan sinyal dengan antarmuka grafis sederhana.

REFERENCES

- [1] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2010.
- [2] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, 4th ed., New York, NY, USA: McGraw-Hill, 2011.
- [3] P. Lathi, *Modern Digital and Analog Communication Systems*, 4th ed., Oxford, U.K.: Oxford Univ. Press, 2018.
- [4] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, Wiley, 1996.
- [5] Python Software Foundation, "Tkinter — Python interface to Tcl/Tk," [Online]. Available: <https://docs.python.org/3/library/tkinter.html>. [Accessed: Sep. 2, 2025].
- [6] SciPy Community, "scipy.signal — Signal Processing," [Online]. Available: <https://docs.scipy.org/doc/scipy/reference/signal.html>. [Accessed: Sep. 2, 2025].
- [7] NumPy Developers, "NumPy Reference," [Online]. Available: <https://numpy.org/doc/stable/>. [Accessed: Sep. 2, 2025].
- [8] Matplotlib Developers, "Matplotlib: Visualization with Python," [Online]. Available: <https://matplotlib.org/stable/index.html>. [Accessed: Sep. 2, 2025].
- [9] J. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 3rd ed., O'Reilly Media, 2022.
- [10] Pydub Documentation, "Manipulate audio with a simple and easy high level interface," [Online]. Available: <https://pydub.com/>. [Accessed: Sep. 2, 2025].