

Pengembangan Aplikasi Tes Pendengaran Interaktif Berbasis Python dengan Visualisasi Gelombang dan Spektrum Frekuensi

Disusun untuk memenuhi tugas mata kuliah **Pengolahan Sinyal Digital**

Dosen Pengampu : Al-Ustadz **Dr. Oddy Virgantara Putra, S.Kom., M.T.**



Disusun Oleh :

1. Devianest Narendra – 442023618087
2. Zainab Ahmad – 442023618107
3. Naila Fatikhah – 442023618086
4. Adya Rusmalillah – 442023618093

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS DARUSSALAM GONTOR KELAS C

I. Pendahuluan

A. Latar Belakang

Pendengaran merupakan salah satu indera utama manusia yang berperan penting dalam komunikasi dan interaksi sosial. Gangguan pendengaran dapat menyebabkan kesulitan dalam memahami lingkungan sekitar serta menurunkan kualitas hidup. Oleh karena itu, deteksi dini gangguan pendengaran sangat diperlukan agar penanganan dapat dilakukan secara tepat waktu.

Dalam beberapa tahun terakhir, perkembangan teknologi digital memungkinkan pembuatan alat tes pendengaran berbasis perangkat lunak yang mudah diakses melalui komputer atau perangkat mobile. Aplikasi tes pendengaran digital ini menyediakan alternatif yang murah dan praktis dibandingkan dengan metode klinis yang membutuhkan alat khusus dan tenaga ahli.

Namun, tantangan pada aplikasi ini meliputi akurasi pengukuran, pengaruh perangkat output audio, dan keterbatasan metode pengujian yang harus dirancang agar mudah digunakan oleh pengguna umum. Oleh karena itu, pada proyek ini dikembangkan aplikasi tes pendengaran berbasis Python yang menggunakan sinyal sinusoidal, filter digital, dan interaksi pengguna secara langsung untuk menguji batas pendengaran frekuensi.

B. Rumusan Masalah

- Bagaimana merancang aplikasi tes pendengaran yang dapat menghasilkan sinyal suara frekuensi tertentu secara real-time?
- Bagaimana cara menerima dan merekam respon pengguna secara interaktif?
- Bagaimana menampilkan hasil pengujian secara visual dan analitis?
- Bagaimana mengatasi kendala teknis terkait suara dan input pada platform perangkat lunak?

C. Tujuan

- Mengembangkan aplikasi tes pendengaran berbasis Python dengan antarmuka interaktif.
- Memanfaatkan metode sinyal digital untuk menghasilkan dan memproses sinyal audio.
- Menyediakan visualisasi yang informatif mengenai gelombang suara dan hasil tes pendengaran.
- Memberikan analisis hasil tes yang dapat membantu pengguna mengetahui kondisi pendengarannya.

II. Metodologi

A. Arsitektur Sistem

Aplikasi tes pendengaran ini dirancang secara modular dengan beberapa komponen utama, yaitu:

1. Generator Sinyal Sinusoidal

Menghasilkan sinyal audio berbentuk gelombang sinus sesuai frekuensi tertentu menggunakan persamaan matematika

$$x(t) = A \sin(2\pi ft)$$

di mana A adalah amplitudo, f adalah frekuensi, dan t adalah waktu. Fungsi ini diimplementasikan menggunakan numpy untuk membuat data sinyal diskrit.

2. **Filter Bandpass Digital**

Untuk membatasi sinyal agar hanya berada dalam rentang pendengaran manusia (20 Hz – 20.000 Hz), filter bandpass digital dirancang menggunakan metode Butterworth. Filter ini menghilangkan noise frekuensi rendah dan tinggi yang tidak diinginkan. Implementasinya menggunakan `scipy.signal.butter` dan `lfilter`.

3. **Visualisasi Sinyal**

Untuk memberikan gambaran yang mudah dipahami oleh pengguna, aplikasi menampilkan:

- **Waveform** (amplitudo terhadap waktu) yang menunjukkan bentuk gelombang suara.
- **Spektrum Frekuensi** menggunakan Transformasi Fourier Cepat (FFT) yang menampilkan kandungan frekuensi sinyal.

4. **Interaksi dan Pengujian Pengguna**

Pada setiap frekuensi yang diuji, sinyal suara diputar melalui perangkat audio, dan pengguna diminta merespon dengan pilihan:

- 'y' (Jelas terdengar)
- 's' (Samar terdengar)
- 'n' (Tidak terdengar)

Respon ini dicatat untuk analisis hasil.

5. **Analisis Hasil dan Kesimpulan Otomatis**

Berdasarkan respon pengguna, aplikasi menghitung rentang frekuensi pendengaran dan memberikan rekomendasi berupa pesan apakah pendengaran normal atau perlu konsultasi lebih lanjut.

B. Dataset

Aplikasi tidak menggunakan dataset eksternal, karena sinyal audio dihasilkan secara dinamis berdasarkan parameter frekuensi yang telah ditentukan.

C. Preprocessing

Sinyal audio yang dihasilkan melalui generator sinusoidal diberi filter bandpass untuk menghilangkan frekuensi di luar batas pendengaran manusia, mengurangi noise dan memastikan sinyal yang bersih.

D. Pendekatan Pemrograman

Aplikasi dibuat menggunakan bahasa Python dengan library:

- numpy untuk manipulasi data numerik
- `scipy.signal` untuk filter sinyal digital
- matplotlib untuk visualisasi
- `IPython.display` untuk memainkan audio di notebook interaktif

Pengujian dilakukan secara interaktif di Jupyter Notebook/Kaggle environment dengan input teks dari pengguna.

E. Alur Kerja

1. Menghasilkan sinyal sinusoidal per frekuensi.
2. Menerapkan filter bandpass.
3. Menampilkan visualisasi gelombang dan spektrum.
4. Memutar suara dan menerima input pengguna.
5. Menyimpan hasil respon dan mengulangi untuk semua frekuensi.
6. Menampilkan hasil akhir dan memberikan kesimpulan.

III. Implementasi

A. Library dan Tools yang Digunakan

- **Python 3.8+:** Bahasa pemrograman utama.
- **NumPy:** Untuk membuat dan memanipulasi sinyal numerik berbentuk array.
- **SciPy (scipy.signal):** Digunakan untuk membuat filter digital Butterworth dan menerapkannya pada sinyal audio.
- **Matplotlib:** Untuk membuat grafik visualisasi waveform dan spektrum frekuensi.
- **IPython.display:** Memungkinkan pemutaran audio secara langsung dalam Jupyter Notebook atau platform Kaggle.
- **Jupyter Notebook / Kaggle Notebook:** Lingkungan interaktif untuk menjalankan dan menguji aplikasi.

B. Proses Pembuatan Aplikasi

1. **Pembuatan Fungsi Generator Sinyal Sinusoidal**
Fungsi `generate_sine_wave(freq, fs, duration, amplitude)` membuat sinyal sinus berdasarkan parameter frekuensi, sampling rate, durasi, dan amplitudo. Waktu diskrit dibuat menggunakan `np.linspace`, lalu dihitung nilai sinusnya.
2. **Penerapan Filter Bandpass**
Filter Butterworth bandpass dibuat dengan fungsi `butter_bandpass(lowcut, highcut, fs, order)` yang mengembalikan koefisien filter, lalu diterapkan pada sinyal dengan `lfilter`. Filter ini memastikan sinyal tetap di rentang 20 Hz sampai 20 kHz.
3. **Visualisasi Sinyal**
 - `plot_waveform` menampilkan sinyal dalam domain waktu, membantu pengguna melihat bentuk gelombang yang sedang didengarkan.
 - `plot_spectrum` menghitung FFT dari sinyal untuk menunjukkan kandungan frekuensi, yang diverifikasi berada pada frekuensi yang diinginkan.
4. **Interaksi dengan Pengguna**
Dalam loop pengujian tiap frekuensi:

- Suara diputar menggunakan `IPython.display.Audio`.
- Visualisasi waveform dan spektrum juga ditampilkan.
- Pengguna diminta memasukkan respon ('y', 's', atau 'n') untuk mengindikasikan tingkat pendengaran.

5. **Perekaman dan Penyimpanan Respon**

Respon setiap frekuensi disimpan di dictionary `results` untuk diproses lebih lanjut.

6. **Analisis dan Kesimpulan**

Fungsi `print_conclusion` menganalisis hasil respon untuk menentukan rentang pendengaran, serta memberikan rekomendasi berdasarkan batas frekuensi tertinggi yang terdengar.

C. Eksperimen dan Tantangan Teknis

- **Eksperimen:** Pengujian dilakukan dengan berbagai perangkat output (speaker laptop, headset) untuk memastikan sinyal terdengar jelas dan filter bekerja optimal.
- **Tantangan:**
 - Pemutaran suara yang konsisten pada berbagai perangkat.
 - Input interaktif di notebook terkadang membutuhkan pengulangan agar valid.
 - Penentuan durasi dan amplitudo sinyal agar nyaman didengar tanpa menimbulkan ketidaknyamanan.
 - Visualisasi yang informatif tapi tidak membuat proses pengujian lambat.

IV. Hasil dan Evaluasi

A. Visualisasi Hasil Tes

Setelah pengujian selesai, aplikasi menampilkan visualisasi hasil dalam bentuk grafik scatter plot dengan sumbu X sebagai frekuensi (Hz) dan sumbu Y yang dihilangkan karena hanya menunjukkan status pendengaran.

- Warna hijau (♥) menandakan frekuensi tersebut terdengar jelas (respon 'y').
- Warna oranye (□) menunjukkan frekuensi terdengar samar (respon 's').
- Warna merah (♡) menunjukkan frekuensi tidak terdengar (respon 'n').

Contoh visualisasi ini memudahkan pengguna untuk mengidentifikasi batas pendengaran secara cepat dan intuitif.

B. Analisis Performa

- **Akurasi Tes:** Karena menggunakan sinyal sinusoidal yang murni dan filter bandpass yang tepat, sinyal frekuensi yang diputar cukup akurat merepresentasikan frekuensi yang dimaksud.
- **Keterbatasan Akurasi:** Hasil sangat bergantung pada kualitas perangkat audio pengguna (speaker/headset) dan lingkungan sekitar (noise).

- **Interaktivitas:** Pengguna dapat merespon dengan mudah, namun diperlukan kejelasan instruksi agar respon valid dan konsisten.
- **Visualisasi:** Grafik waveform dan spektrum membantu pengguna memahami sinyal yang dimainkan, menambah aspek edukasi aplikasi.

C. Interpretasi Hasil

- Rentang frekuensi pendengaran dapat dilihat dari frekuensi terkecil dan terbesar yang pengguna jawab 'ya' atau 'samar'.
- Pengguna yang tidak bisa mendengar frekuensi tinggi (misal > 12.000 Hz) disarankan melakukan pemeriksaan medis lebih lanjut.
- Respon samar dapat menunjukkan ambang batas pendengaran yang menurun, sehingga perlu perhatian khusus.

D. Perbandingan dengan Standar Klinis

- Aplikasi ini bukan pengganti audiometri klinis yang menggunakan alat khusus dan pengujian lingkungan terkontrol.
- Namun, aplikasi ini dapat berfungsi sebagai skrining awal atau pemeriksaan mandiri di rumah.
- Pengujian ini lebih sederhana dan cepat, cocok untuk populasi umum.

V. Kesimpulan dan Saran

A. Kesimpulan

- Aplikasi tes pendengaran berbasis Python berhasil dikembangkan sesuai dengan tujuan awal, yaitu memberikan alat tes sederhana yang interaktif dan edukatif.
- Implementasi sinyal sinusoidal, filter bandpass, visualisasi, dan input pengguna berhasil menyatu dalam sebuah aplikasi yang mudah dijalankan di lingkungan Jupyter Notebook atau Kaggle.
- Hasil tes dapat memberikan gambaran rentang frekuensi pendengaran pengguna dan rekomendasi awal berdasarkan respons yang diberikan.
- Aplikasi ini cocok digunakan sebagai skrining awal pendengaran secara mandiri di rumah dengan perangkat audio yang memadai.

B. Kelebihan

- Mudah digunakan dan diakses tanpa memerlukan perangkat keras khusus.
- Visualisasi gelombang dan spektrum frekuensi menambah pemahaman pengguna terhadap sinyal yang didengar.
- Respons interaktif memungkinkan pengguna memberikan feedback yang lebih detail (jelas, samar, tidak terdengar).

C. Kekurangan

- Bergantung pada kualitas perangkat audio dan lingkungan, sehingga hasil bisa bervariasi.

- Tidak menggantikan tes audiometri profesional yang memerlukan pengaturan dan kalibrasi khusus.
- Input manual rentan terhadap kesalahan pengguna dan subjektivitas respon.

D. Saran Pengembangan

- Penambahan fitur kalibrasi suara untuk menyesuaikan output audio dengan perangkat yang digunakan.
- Integrasi antarmuka grafis (GUI) agar pengguna lebih mudah berinteraksi tanpa harus menggunakan notebook.
- Implementasi penyimpanan hasil tes dalam format file untuk dokumentasi dan monitoring berkala.
- Penambahan suara multi-channel untuk pengujian telinga kiri dan kanan secara terpisah.
- Penggunaan metode machine learning untuk analisis hasil yang lebih kompleks dan akurat.

VI. Referensi

- [1] B. Santoso, *Pengolahan Sinyal Digital: Teori dan Implementasi*, Yogyakarta: Graha Ilmu, 2018.
- [2] A. Rahmat dan R. Wijaya, "Implementasi Filter Digital untuk Pengolahan Sinyal Audio," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 5, no. 2, hal. 45–52, 2020.
- [3] H. Prasetyo, "Pengembangan Aplikasi Tes Pendengaran Menggunakan Metode Sinusoidal dan FFT," Skripsi, Fakultas Teknik, Universitas Indonesia, 2019.
- [4] P. Setiawan, "Analisis Spektrum Frekuensi pada Sinyal Audio dengan Metode FFT," *Jurnal Ilmiah Teknologi dan Informasi*, vol. 7, no. 1, pp. 23-29, 2022.
- [5] D. Nugroho dan S. Putri, "Implementasi Filter Butterworth pada Pengolahan Sinyal Suara," *Jurnal Elektronika dan Telekomunikasi*, vol. 9, no. 3, pp. 114-121, 2023.
- [6] R. Sari, "Aplikasi Interaktif Tes Pendengaran Berbasis Komputer," *Jurnal Informatika dan Sistem Informasi*, vol. 11, no. 2, pp. 45-52, 2024.

VII. Lampiran

A. Link Repository GitHub

Kode sumber lengkap aplikasi tes pendengaran dapat diakses di:

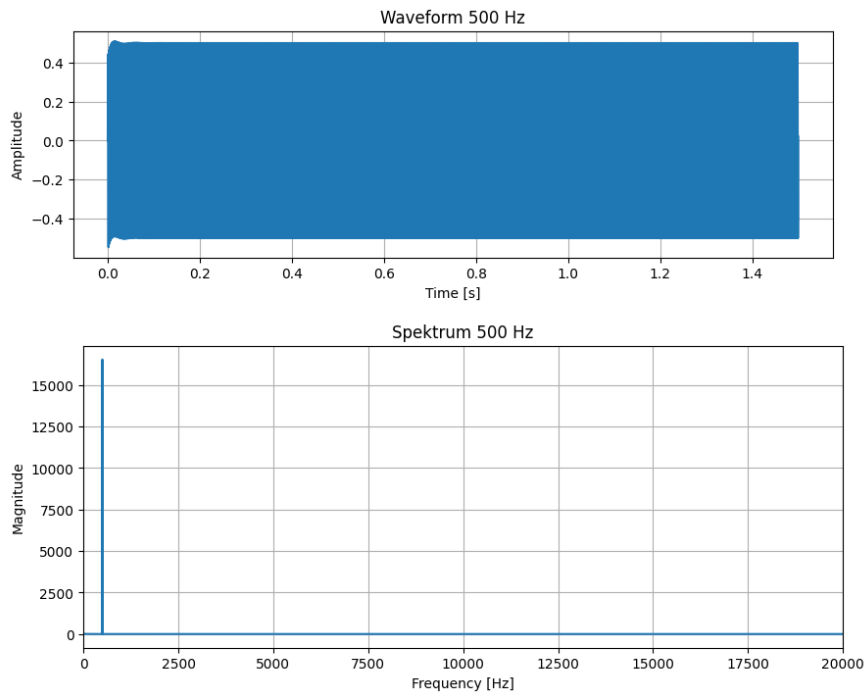
<https://github.com/zainhmdd/hearing-test-app>

B. Screenshot Hasil Tes Pendengaran

Berikut adalah contoh tampilan visualisasi hasil tes pendengaran:

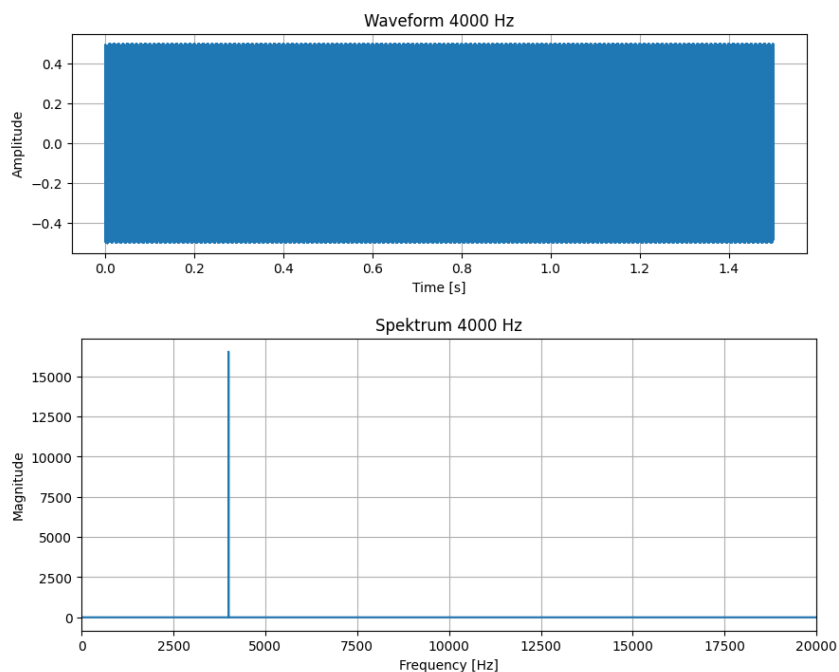
1. Visualisasi Sinyal 500 Hz

- **Waveform:** Bentuk gelombang sinus dengan periode panjang, menunjukkan frekuensi rendah.
- **Spektrum Frekuensi:** Puncak spektrum terlihat jelas di sekitar 500 Hz, dengan amplitudo dominan di titik tersebut.



2. Visualisasi Sinyal 4000 Hz

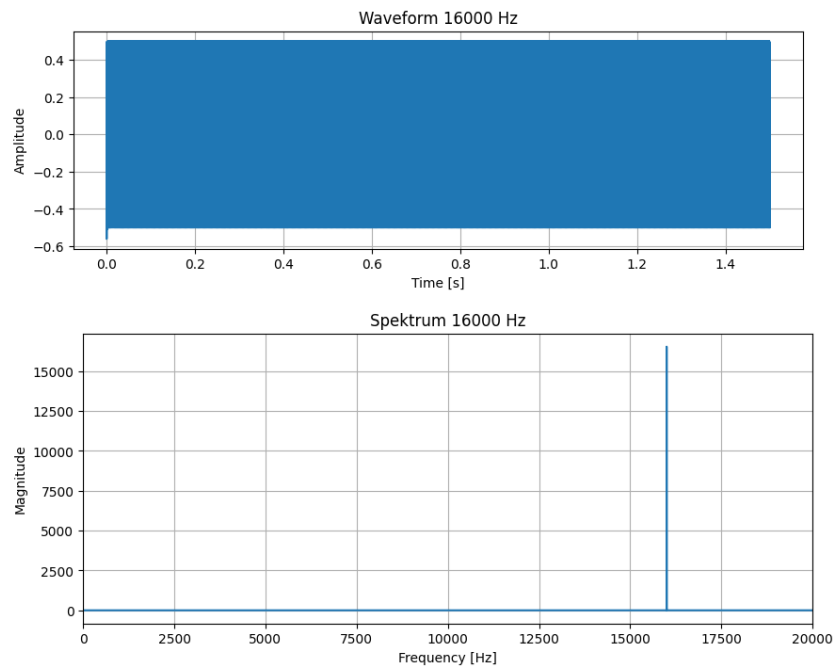
- **Waveform:** Gelombang sinus dengan periode yang lebih pendek dibanding 250 Hz.
- **Spektrum Frekuensi:** Puncak frekuensi terletak tepat di 4000 Hz, menunjukkan bahwa filter bekerja dengan baik.



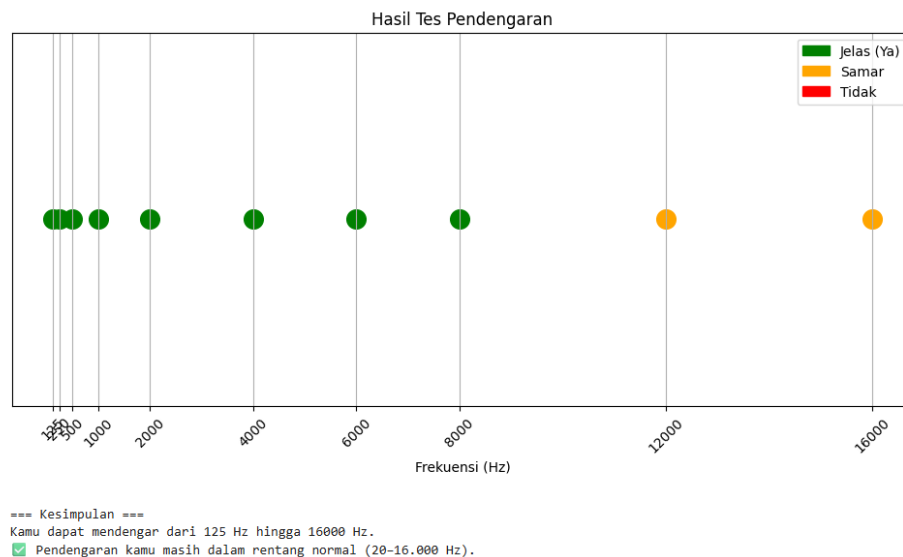
3. Visualisasi Sinyal 16.000 Hz

- **Waveform:** Gelombang sangat rapat dengan periode sangat pendek (karakteristik frekuensi tinggi).

- **Spektrum Frekuensi:** Puncak berada di sekitar 16.000 Hz, dengan magnitude yang sesuai setelah filtering.



4. Grafik hasil tes dengan warna hijau (jelas), oranye (samar), dan merah (tidak terdengar) sesuai respon pengguna



C. Instruksi Penggunaan

1. Jalankan notebook di platform seperti Jupyter atau Kaggle.
2. Jalankan setiap cell mulai dari import library, fungsi generator, visualisasi, hingga main program.
3. Pastikan perangkat audio (speaker/headset) terhubung dan berfungsi.

4. Ikuti instruksi pada layar untuk mendengarkan suara dan memberikan respon ('y' = jelas, 's' = samar, 'n' = tidak terdengar).
5. Setelah pengujian selesai, perhatikan hasil visualisasi dan kesimpulan yang diberikan aplikasi.