

# Transformasi Fourier: Teori Dasar dan Aplikasinya dalam Penghapusan Noise pada Audio

<nama-mahasiswa>  
NRP: <nrp-mahasiswa>

15 November 2025

Disusun untuk tugas makalah Quiz 2 mata kuliah Pengantar Teori Aproksimasi

## Ringkasan

Makalah ini membahas penerapan transformasi Fourier untuk penghapusan noise pada sinyal audio. Dimulai dari landasan teori transformasi Fourier kontinu/discret beserta sifat-sifat kuncinya, dilanjutkan dengan metodologi *spectral gating* dan *spectral subtraction* pada domain frekuensi, termasuk perumusan filter Wiener. Implementasi dilakukan menggunakan dua pendekatan: fitur *Noise Reduction* pada Audacity serta skrip Python berbasis NumPy–SciPy dan librosa. Hasil menunjukkan bahwa pemodelan spektrum noise dari *noise profile* dan penapisan pada domain frekuensi mampu menurunkan tingkat noise tanpa menurunkan kejernihan ujaran secara signifikan jika parameter ambang dipilih hati-hati.<sup>1</sup>

## 1 Pendahuluan

Transformasi Fourier (TF) memetakan sinyal waktu ke domain frekuensi untuk menganalisis komponen harmoniknya. Untuk fungsi terintegralkan absolut  $f \in L^1(\mathbb{R})$ , TF didefinisikan sebagai

$$\mathcal{F}\{f\}(\xi) = \hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx, \quad (1.1)$$

dan inversnya diberikan oleh

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i x \xi} d\xi, \quad (1.2)$$

di bawah syarat reguler yang sesuai[1, 2]. Pada praktik komputasi, digunakan transformasi Fourier diskret (DFT) yang dievaluasi efisien dengan FFT[3].

Pada pemrosesan audio, banyak jenis gangguan (*hiss, hum, broadband noise*) menumpang pada sinyal ujaran. Dengan memanfaatkan representasi spektral melalui TF/DFT, kita dapat memisahkan komponen noise dan sinyal lalu menerapkan penapisan selektif untuk mereduksi noise tanpa memodifikasi struktur temporal secara drastis[1].

## 2 Sifat-sifat Transformasi Fourier yang Relevan

Berikut sifat-sifat pokok yang digunakan pada perancangan penapis di domain frekuensi[2]:

- Linearitas:  $\mathcal{F}\{af + bg\} = a\hat{f} + b\hat{g}$ .
- Pergeseran waktu:  $\mathcal{F}\{f(t - t_0)\} = e^{-2\pi i \xi t_0} \hat{f}(\xi)$ .

---

<sup>1</sup>Kode contoh dan berkas proyek dapat ditaruh pada lampiran atau repositori terpisah bila diperlukan.

- Teorema konvolusi:  $\mathcal{F}\{(f * g)(t)\} = \hat{f}(\xi) \hat{g}(\xi)$ .
- Parseval/Plancherel:  $\int |f(t)|^2 dt = \int |\hat{f}(\xi)|^2 d\xi$ .

Konvolusi pada waktu menjadi perkalian pada frekuensi memungkinkan perancangan filter sebagai fungsi transfer  $H(\xi)$  yang mengalikan spektrum sinyal:  $\hat{y}(\xi) = H(\xi) \hat{x}(\xi)$ .

### 3 Landasan Teori Denoising Audio

#### 3.1 STFT dan Spektrogram

Untuk sinyal tidak tunak, digunakan *short-time Fourier transform* (STFT) dengan jendela  $w[1]$ :

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t - \tau) e^{-i\omega t} dt. \quad (3.1)$$

Magnitudo  $|X(\tau, \omega)|$  menghasilkan spektrogram. Denoising berbasis spektrum biasanya memodifikasi magnitudo lalu merekonstruksi sinyal menggunakan invers STFT.

#### 3.2 Pengurangan Spektrum dan *Spectral Gating*

Misalkan  $y(t) = s(t) + n(t)$  adalah sinyal terkontaminasi noise. Pada domain frekuensi diperoleh  $Y = S + N$ . *Spectral subtraction* memperkirakan magnitudo sinyal bersih dengan

$$|\hat{S}(\xi)| = \max \{ |Y(\xi)| - \alpha \hat{\sigma}_N(\xi), 0 \}, \quad (3.2)$$

di mana  $\hat{\sigma}_N(\xi)$  adalah estimasi magnitudo noise (diukur dari segmen *noise-only*) dan  $\alpha > 1$  adalah faktor *over-subtraction*[4]. *Spectral gating* mengatur ambang pada tiap-bin frekuensi dan meneduhkan (mengecilkan) koefisien yang diperkirakan berasal dari noise.

#### 3.3 Filter Wiener

Dalam kerangka kuadrat-terkecil, filter Wiener meminimalkan MSE antara keluaran dan sinyal bersih. Pada domain frekuensi kontinu berlaku[5]:

$$H(\xi) = \frac{S_{xx}(\xi)}{S_{xx}(\xi) + S_{nn}(\xi)}, \quad (3.3)$$

di mana  $S_{xx}$  dan  $S_{nn}$  adalah densitas spektral daya sinyal dan noise. Dalam praktik, rasio  $\text{SNR}(\xi)$  diestimasi dari spektrogram sehingga  $H(\xi) = \frac{\text{SNR}(\xi)}{\text{SNR}(\xi)+1}$ .

## 4 Metodologi dan Perangkat

Kami mendemonstrasikan dua alur kerja berikut.

#### 4.1 Audacity: Noise Reduction

Audacity menyediakan efek *Noise Reduction* yang mengimplementasikan penapisan berbasis profil noise dan ambang spektral[6]. Langkah-langkah:

1. Pilih cuplikan audio yang hanya berisi noise (*noise profile*).
2. Effects → Noise Reduction → Get Noise Profile.
3. Seleksi seluruh audio, atur parameter: *Reduction* (dB), *Sensitivity*, dan *Frequency smoothing* (bands).
4. Terapkan dan dengarkan hasil; ulangi penyesuaian hingga artefak minimal.

## 4.2 Python: NumPy/SciPy dan librosa

Implementasi komputasional memakai STFT, pengurangan spektrum, dan opsi filter Wiener. Pustaka yang digunakan: NumPy[7], SciPy[8], dan librosa[9]. Garis besar algoritma:

1. Baca sinyal  $y[n]$  (mono,  $f_s$  Hz). Hitung STFT:  $Y[k, m]$ .
2. Estimasi spektrum noise  $\hat{\sigma}_N[k]$  dari segmen hening.
3. Terapkan (3.2) per *bin* (atau gunakan bentuk Wiener (3.3)).
4. Rekonstruksi fase dari  $Y[k, m]$  (fase asli) dan lakukan invers STFT.
5. Simpan hasil  $\hat{s}[n]$  sebagai WAV.

## 5 Eksperimen

Data uji berupa rekaman ujaran dengan *broadband noise* kipas. Parameter tipikal yang bekerja baik: panjang jendela 1024 sampel, hop 256, jendela Hann; faktor  $\alpha$  antara 1.5–2.0; penghalusan frekuensi 3–6 pita pada Audacity. Evaluasi kualitatif dilakukan lewat dengar-subjektif serta metrik sederhana seperti *segmental SNR* sebelum dan sesudah pemrosesan[1].

## 6 Hasil dan Pembahasan

Pengurangan spektrum efektif menekan dengung dan desis stasioner. Namun nilai  $\alpha$  terlalu besar memicu artefak *musical noise*. Filter Wiener cenderung menghasilkan hasil lebih halus dengan distorsi lebih kecil saat estimasi SNR memadai, tetapi kurang agresif untuk noise berat. Pada Audacity, kombinasi *Sensitivity* sedang dan *Frequency smoothing* lebih besar membantu mengurangi *musical noise*. Pada pipeline Python, penggunaan *soft mask* berbasis SNR membuat transisi antar-waktu lebih mulus.

## 7 Kesimpulan

Transformasi Fourier memungkinkan denoising audio yang efektif melalui manipulasi magnitudo spektrum pada domain frekuensi. Dua jalur Audacity dan implementasi Python menunjukkan bahwa pemodelan profil noise dan penapisan berbasis ambang/Wiener dapat meningkatkan SNR dan kejernihan ujaran bila parameter disetel dengan hati-hati. Pekerjaan lanjut dapat mencakup pemodelan adaptif SNR per waktu-frekuensi dan pendekatan pembelajaran mendalam yang tetap berlandas pada representasi STFT.

## Pustaka

- [1] A. V. Oppenheim **and** R. W. Schafer, *Discrete-Time Signal Processing*, 3 **edition**. Pearson, 2010.
- [2] R. N. Bracewell, *The Fourier Transform and Its Applications*, 3 **edition**. McGraw-Hill, 2000.
- [3] J. W. Cooley **and** J. W. Tukey, ?An Algorithm for the Machine Calculation of Complex Fourier Series,? *Mathematics of Computation*, **jourvol** 19, **number** 90, **pages** 297–301, 1965.

- [4] S. F. Boll, ?Suppression of Acoustic Noise in Speech Using Spectral Subtraction,? *IE-EE Transactions on Acoustics, Speech, and Signal Processing*, **jourvol** 27, **number** 2, **pages** 113–120, 1979.
- [5] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. MIT Press, 1949.
- [6] Audacity Team. ?Noise Reduction,? **urlseen** 15 november 2025. **url:** <https://support.audacityteam.org/repair/noise-reduction>
- [7] C. R. Harris **and others**, ?Array programming with NumPy,? *Nature*, **jourvol** 585, **pages** 357–362, 2020.
- [8] P. Virtanen **and others**, ?SciPy 1.0: fundamental algorithms for scientific computing in Python,? *Nature Methods*, **jourvol** 17, **pages** 261–272, 2020.
- [9] B. McFee **and others**, ?librosa: Audio and Music Signal Analysis in Python,? **in** *Proceedings of the 14th Python in Science Conference* K. Huff **and** J. Bergstra, **editors**, 2015, **pages** 18–24.