

# Identifikasi dan Klasifikasi Suara Vokal Menggunakan Metode *Fast Fourier Transform*

I Ketut Manik Ambarawan<sup>a1</sup>, I Gusti Agung Gede Arya Kadyanan<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>ambarawan.2308561017@student.unud.ac.id  
<sup>2</sup>gungde@unud.ac.id

## Abstract

*This research develops a vocal range identification and classification system using frequency spectrum analysis and Fast Fourier Transform (FFT) algorithm. The system addresses the need for an accessible vocal range identification tool for amateur singers and the public without formal musical training. The proposed system combines real-time audio recording capabilities with audio file processing, implementing pitch identification through FFT analysis and windowing functions. The system features two input methods: real-time recording and audio file upload supporting various formats (MP3, WAV, FLAC, AAC, OGG, M4A). Using PyAudio for real-time processing and Librosa for file analysis, the system accurately identifies fundamental frequencies within the human vocal range (80-1100 Hz) and automatically classifies voice types (Bass, Baritone, Tenor, Alto, Mezzo-soprano, Soprano). Testing demonstrates effective frequency identification with pitch conversion accuracy ranging from 95.7% to 98.8% and voice type classification achieving 81.2% accuracy. The system provides an efficient solution for vocal range analysis with low computational complexity and real-time processing capabilities.*

**Keywords:** *Vocal Range Identification, FFT Algorithm, Pitch Identification, Voice Classification, Audio Processing, Frequency Analysis*

## 1. Pendahuluan

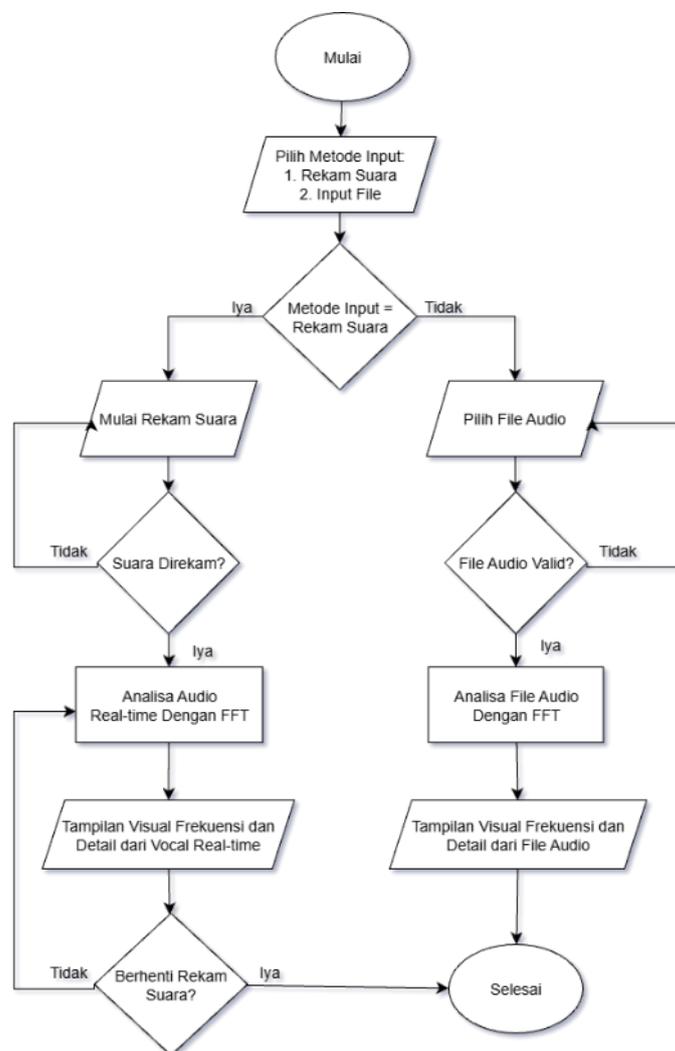
Musik merupakan bahasa universal yang dapat dinikmati oleh semua kalangan masyarakat dan menjadi *medium* ekspresi yang sangat kuat [1]. Pemahaman tentang ambitus vokal atau rentang suara yang dapat dicapai oleh seorang penyanyi merupakan aspek fundamental yang sangat penting dalam dunia musik vokal [2]. Namun, sebagian besar masyarakat awam mengalami kesulitan dalam mengidentifikasi ambitus vokal mereka sendiri karena keterbatasan pengetahuan musik formal dan akses terhadap pelatih vokal profesional. Di era digital saat ini, platform media sosial seperti YouTube, TikTok, dan aplikasi karaoke *online* telah mendorong minat masyarakat umum untuk mengekspresikan diri melalui bernyanyi [3]. Ketidaktahuan tentang ambitus vokal dapat menyebabkan pemilihan lagu yang tidak sesuai dengan kemampuan vokal, penurunan kualitas suara, dan risiko cedera pada pita suara akibat memaksakan produksi nada yang berada di luar kemampuan alami [2]. Penelitian sebelumnya dalam bidang identifikasi vokal telah mengeksplorasi berbagai pendekatan teknologi. Sun J et al. [4] melakukan analisis vokal berbasis spektrum gambar, sedangkan Zhang et al. [5] mengembangkan sistem menggunakan *Long-Term Recurrent Convolutional Network* dengan akurasi tinggi namun memerlukan komputasi kompleks. Meskipun menunjukkan kemajuan signifikan, sebagian besar penelitian masih berfokus pada optimasi algoritma tanpa mempertimbangkan kemudahan penggunaan bagi masyarakat awam. Algoritma *Fast Fourier Transform (FFT)* telah terbukti efektif dalam analisis frekuensi sinyal audio dan identifikasi pitch dalam berbagai aplikasi musik [6]. *FFT* memberikan keunggulan dalam hal efisiensi komputasi dan akurasi identifikasi frekuensi yang cocok untuk implementasi dalam aplikasi desktop yang dapat digunakan oleh masyarakat umum. Penelitian ini bertujuan mengembangkan sistem identifikasi dan klasifikasi ambitus vokal yang menggabungkan analisis spektrum frekuensi dengan algoritma *FFT* yang efisien. Sistem dirancang untuk memberikan solusi yang akurat dan praktis bagi identifikasi karakteristik vokal

melalui dua metode input, yaitu perekaman langsung menggunakan mikrofon dan analisis *file* audio. Hasil yang diharapkan adalah sistem yang dapat mengidentifikasi rentang vokal dengan akurasi tinggi, mengklasifikasikan jenis suara, dan memberikan analisis frekuensi yang informatif.

## 2. Metode Penelitian

### 2.1. Deskripsi Sistem

Sistem identifikasi ambitus vokal dikembangkan menggunakan pendekatan pemrosesan sinyal digital dengan implementasi algoritma *Fast Fourier Transform* untuk analisis spektrum frekuensi. Sistem ini dirancang dengan arsitektur modular yang memungkinkan fleksibilitas dalam pengembangan dan optimasi [7]. Arsitektur sistem terdiri dari tiga komponen utama: modul akuisisi audio yang bertanggung jawab untuk perekaman dan pembacaan file audio, modul pemrosesan sinyal yang mengimplementasikan algoritma *FFT* untuk analisis frekuensi, dan modul klasifikasi yang menentukan jenis suara berdasarkan karakteristik frekuensi yang teridentifikasi. *Flowchart* sistem dapat dilihat pada Gambar 1.



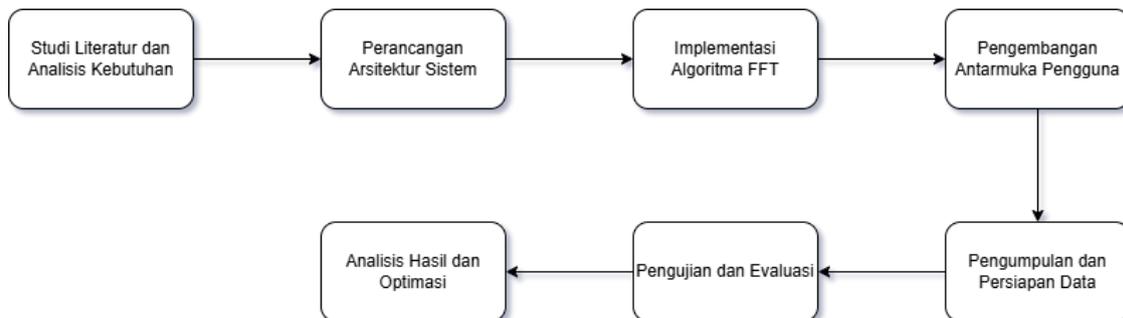
**Gambar 1.** *Flowchart* Sistem Identifikasi dan Klasifikasi Ambitus Vokal

Berdasarkan *flowchart* pada Gambar 1, sistem identifikasi ambitus vokal menyediakan dua metode input yaitu rekam suara *real-time* dan input file audio, dimana kedua metode tersebut akan divalidasi terlebih dahulu sebelum dianalisis menggunakan algoritma *FFT* untuk

menghasilkan tampilan visual frekuensi dan detail vokal, kemudian sistem memberikan kontrol kepada pengguna untuk menghentikan atau melanjutkan proses hingga analisis selesai.

## 2.2. Tahapan Penelitian

Penelitian ini dilaksanakan melalui beberapa tahapan sistematis yang dirancang untuk menghasilkan sistem yang optimal dan dapat diandalkan. Tahapan penelitian bisa dilihat pada Gambar 2.



**Gambar 2.** Tahapan Penelitian

Penelitian ini dilakukan melalui tujuh tahap sistematis yang dimulai dengan studi literatur dan analisis kebutuhan untuk memahami penelitian sebelumnya serta kebutuhan pengguna, dilanjutkan dengan perancangan arsitektur sistem dan implementasi algoritma *Fast Fourier Transform (FFT)* untuk analisis spektrum frekuensi [6]. Tahap selanjutnya mencakup pengembangan antarmuka pengguna grafis menggunakan *Tkinter*, persiapan *dataset* yang terdiri dari data sintesis dan vokal nyata, serta pengujian komprehensif untuk mengevaluasi akurasi identifikasi frekuensi dan klasifikasi jenis suara. Penelitian diakhiri dengan analisis hasil dan optimasi sistem berdasarkan temuan yang diperoleh dari pengujian.

## 2.3. Pengumpulan Data

Data yang digunakan dalam penelitian ini diklasifikasikan ke dalam dua kategori utama, yaitu data untuk pengujian akurasi identifikasi frekuensi dan data untuk pengujian klasifikasi jenis suara. Masing-masing kategori dirancang untuk mendukung kebutuhan evaluasi sistem berdasarkan jenis pengujian yang dilakukan.

### a. Data untuk Pengujian Akurasi Identifikasi Frekuensi

Untuk menguji akurasi identifikasi frekuensi sistem, digunakan *tone generator* (<https://onlinetonegenerator.com>) untuk menghasilkan sinyal audio sintesis dengan frekuensi yang telah ditentukan secara presisi. Tabel 1 menunjukkan detail data yang digunakan.

**Tabel 1.** Data untuk Pengujian Akurasi Identifikasi Frekuensi

Nama File	Frekuensi	Format	Durasi	Ukuran File	Sample Rate
80.wav	80 Hz	WAV	10 detik	1.723 KB	44.100 Hz
200.wav	200 Hz	WAV	10 detik	1.723 KB	44.100 Hz
400.wav	400 Hz	WAV	10 detik	1.723 KB	44.100 Hz
600.wav	600 Hz	WAV	10 detik	1.723 KB	44.100 Hz
800.wav	800 Hz	WAV	10 detik	1.723 KB	44.100 Hz
1100.wav	1100 Hz	WAV	10 detik	1.723 KB	44.100 Hz

Semua file memiliki durasi yang konsisten untuk memastikan kondisi pengujian yang seragam.

**b. Data untuk Pengujian Klasifikasi Jenis Suara**

Untuk pengujian klasifikasi jenis suara, digunakan dataset *VocalSet: A Singing Voice Dataset*. Dataset ini merupakan kumpulan rekaman vokal berkualitas tinggi yang terdiri dari 10,1 jam rekaman audio dari 20 penyanyi profesional (11 pria, 9 wanita) yang mendemonstrasikan 17 teknik vokal yang berbeda. Dataset ini berisi rekaman audio monofonik dari penyanyi profesional yang mendemonstrasikan teknik vokal standar dan diperluas pada semua 5 vokal [8].

Dalam penelitian ini, digunakan subset dari dataset *VocalSet* yang terdiri dari 30 file audio dengan 5 file per jenis suara seperti yang ditunjukkan pada Tabel 2. Dataset ini dipilih karena kualitas rekamannya yang tinggi dan klasifikasi jenis suara yang sudah tervalidasi oleh para ahli.

**Tabel 2.** Data untuk Pengujian Klasifikasi Jenis Suara

Jenis Suara	Jumlah File	Durasi	Sample Rate	Nama File
Bass	5	10–12 detik	44.100 Hz	f1_scales_c_slow_forte_a.wav hingga f5_scales_c_slow_forte_a.wav
Bariton	5	10–12 detik	44.100 Hz	f6_scales_c_slow_forte_a.wav hingga f10_scales_c_slow_forte_a.wav
Tenor	5	10–12 detik	44.100 Hz	f11_scales_c_slow_forte_a.wav hingga f15_scales_c_slow_forte_a.wav
Alto	5	10–12 detik	44.100 Hz	m1_scales_c_slow_forte_a.wav hingga m5_scales_c_slow_forte_a.wav
Mezzo-soprano	5	10–12 detik	44.100 Hz	m6_scales_c_slow_forte_a.wav hingga m10_scales_c_slow_forte_a.wav
Soprano	5	10–12 detik	44.100 Hz	m11_scales_c_slow_forte_a.wav hingga m15_scales_c_slow_forte_a.wav

**2.4. Algoritma Fast Fourier Transform**

*Fast Fourier Transform (FFT)* merupakan metode komputasi matematis yang berfungsi mengkonversi sinyal analog ke dalam bentuk sinyal digital berbasis spektrum frekuensi [6]. Algoritma ini memecah suatu sinyal menjadi komponen-komponen frekuensi yang beragam menggunakan fungsi eksponensial kompleks. *FFT* adalah implementasi cepat dan efektif dari algoritma *Discrete Fourier Transform* untuk menghitung transformasi *Fourier* secara diskrit. Mengingat sifat kontinyu dari sinyal-sinyal dalam sistem komunikasi, hasil perhitungan ini dapat diaplikasikan untuk transformasi *Fourier* [9].

Transformasi Fourier dapat diekspresikan melalui persamaan:

$$S(f) = \int s(t) \times e^{-j2\pi ft} dt \tag{1}$$

Di mana:

- $S(f)$  mewakili transformasi Fourier dari  $S(t)$ .
- $e$  adalah basis logaritma natural.
- $j$  adalah satuan imajiner (dengan  $j^2 = -1$ ).

Berdasarkan persamaan integral tersebut, *FFT* memungkinkan perhitungan nilai frekuensi, amplitudo, dan fase dari suatu gelombang sinyal. Untuk komputasi spektrum frekuensi sinyal pada sistem digital, diperlukan algoritma *Discrete Fourier Transform (DFT)* yang mengkonversi sinyal ranah waktu menjadi sinyal ranah frekuensi [9].

Formula DFT adalah:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp \left[ \frac{-2j\pi ux}{N} \right] \quad (2)$$

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \left( \cos \left( \frac{2j\pi ux}{N} \right) - j \sin \left( \frac{2j\pi ux}{N} \right) \right) \quad (3)$$

Dengan N adalah total sampel yang diambil.

Proses identifikasi *pitch* menggunakan *FFT* meliputi tahapan preprocessing audio dengan *windowing*, *hanning* untuk mengurangi *spectral leakage*, transformasi *FFT* untuk konversi *domain*, dan identifikasi puncak frekuensi untuk identifikasi frekuensi fundamental. Parameter audio yang digunakan sistem dapat dilihat pada Tabel 3.

**Tabel 3.** Parameter Audio Sistem

Parameter	Nilai	Keterangan
Sample Rate	44.1 kHz	Standar audio digital
Chunk Size	4096 samples	Ukuran <i>buffer</i> untuk <i>real-time</i>
Format	Float32	Presisi tinggi untuk analisis
Channels	1 (Mono)	Simplifikasi pemrosesan
Rentang Vokal	80-1100 Hz	Rentang suara manusia

## 2.5. Metode Input Audio

Sistem mendukung dua metode input audio:

### a. Perekaman *Real-time*

Menggunakan pustaka *PyAudio* untuk akuisisi audio langsung dari mikrofon dengan implementasi *callback threading* untuk menghindari *blocking* proses analisis [7]. Metode ini dipilih karena memungkinkan pemrosesan audio secara kontinyu dan *real-time*, memberikan feedback langsung kepada pengguna mengenai karakteristik vokal yang sedang diproduksi. Sistem *buffer circular* diimplementasikan untuk menjaga stabilitas aliran data audio dan mengurangi *latency* pemrosesan.

### b. Analisis File Audio

Menggunakan pustaka *Librosa* [7] untuk memproses berbagai format file audio (MP3, WAV, FLAC, AAC, OGG, M4A) dengan analisis *frame-by-frame*. Pendekatan ini dipilih karena *Librosa* menyediakan algoritma *preprocessing* yang *robust* dan mendukung berbagai format audio. Analisis dilakukan dengan membagi file audio menjadi *frame-frame* dengan *overlaps* 50% untuk memastikan kontinuitas analisis dan mengurangi efek *windowing* pada batas *frame*.

## 2.6. Klasifikasi Tipe Suara

Sistem mengimplementasikan algoritma klasifikasi tipe suara berdasarkan rentang frekuensi yang teridentifikasi sesuai dengan karakteristik vokal standar [10]. Klasifikasi dilakukan dengan memetakan frekuensi minimum dan maksimum yang teridentifikasi ke dalam kategori tipe suara sebagaimana ditunjukkan pada Tabel 4.

**Tabel 4.** Klasifikasi Tipe Suara

Tipe Suara	Frekuensi <i>Min</i>	Frekuensi <i>Max</i>	Rentang <i>Typical</i>
Bass	< 100 Hz	< 350 Hz	E2 - E4

Tipe Suara	Frekuensi <i>Min</i>	Frekuensi <i>Max</i>	Rentang <i>Typical</i>
Baritone	100-150 Hz	< 350 Hz	G2 - G4
Tenor	< 150 Hz	350-700 Hz	C3 - C5
Alto	150-200 Hz	350-700 Hz	G3 - G5
Mezzo-soprano	< 200 Hz	> 700 Hz	A3 - A5
Soprano	> 200 Hz	> 700 Hz	C4 - C6

## 2.7. Implementasi Antarmuka Pengguna

Antarmuka pengguna diimplementasikan menggunakan pustaka *Tkinter* dengan komponen yang terdiri dari panel kontrol untuk pemilihan metode input dan kontrol perekaman, panel status untuk menampilkan informasi *real-time*, panel hasil untuk menampilkan klasifikasi, dan panel visualisasi untuk grafik spektrum frekuensi menggunakan *Matplotlib* [7].

## 2.8. Evaluasi

Evaluasi sistem dilakukan melalui dua metrik utama yaitu akurasi identifikasi frekuensi dan akurasi klasifikasi tipe suara. Untuk akurasi identifikasi frekuensi, digunakan persamaan:

$$Accuracy = \frac{\sum \text{data uji benar klasifikasi}}{\sum \text{total data uji}} \times 100\% \quad (4)$$

$$Mean Error = \frac{1}{N} \sum_{i=1}^n |f_{detected} - f_{reference}| \quad (5)$$

Untuk evaluasi klasifikasi tipe suara, digunakan metrik *precision*, *recall*, dan *F1-score*:

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (8)$$

Keterangan:

- *TP: True Positives* (prediksi benar untuk kelas tertentu)
- *TN: True Negatives* (prediksi benar untuk bukan kelas tersebut)
- *FP: False Positives* (prediksi salah untuk kelas tertentu)
- *FN: False Negatives* (prediksi salah untuk bukan kelas tersebut)

## 3. Hasil dan Pembahasan

### 3.1. Implementasi Sistem

Sistem identifikasi ambitus vokal berhasil diimplementasikan dengan arsitektur modular yang memungkinkan pemrosesan audio secara efisien. Sistem *Python* ini menyediakan dua mode operasi yang dioptimalkan untuk perekaman langsung dan analisis *file* audio. Implementasi algoritma *FFT* menunjukkan performa yang stabil dengan kemampuan pemrosesan *real-time* yang responsif.

### 3.2. Hasil Performa Pemrosesan Audio

Pengujian performa sistem menunjukkan kemampuan dalam mengidentifikasi frekuensi fundamental secara akurat dengan *latency* rendah. Sistem mampu memproses dan menganalisis

spektrum frekuensi dengan efisiensi komputasi yang optimal. Detail performa sistem dapat dilihat pada Tabel 5.

**Tabel 5.** Performa Pemrosesan Audio

<b>Metrik</b>	<b>Nilai</b>	<b>Keterangan</b>
<i>Latency</i>	~10 ms	Waktu pemrosesan per <i>frame</i>
<i>Update Rate</i>	100 Hz	Frekuensi update visualisasi
<i>CPU Usage</i>	2-5%	Pada sistem Ryzen 7 8845HS
<i>Memory Usage</i>	80-100 MB	Penggunaan RAM puncak

Berdasarkan hasil pengujian pada Tabel 5, sistem menunjukkan performa yang sangat baik dengan latency 10 milidetik dan update rate 100 Hz yang memberikan responsivitas tinggi dan visualisasi spektrum frekuensi yang halus. Penggunaan *CPU* yang rendah (2-5%) dan konsumsi memori yang stabil (80-100 MB) menunjukkan efisiensi algoritma *FFT* dan manajemen buffer audio yang optimal, memungkinkan sistem berjalan lancar tanpa membebani sumber daya sistem.

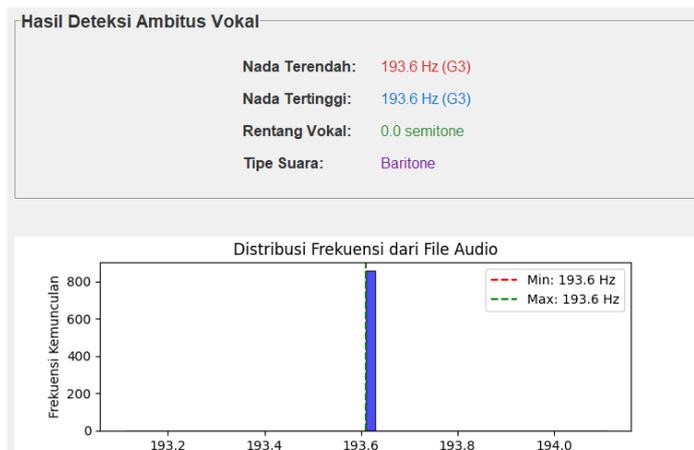
### 3.3. Hasil Evaluasi Akurasi Identifikasi Frekuensi

Pengujian akurasi identifikasi frekuensi dilakukan menggunakan *tone generator* dengan frekuensi referensi yang diketahui dalam rentang vokal manusia (80-1100 Hz). Hasil pengujian menunjukkan variasi akurasi *across different frequency ranges* seperti yang ditunjukkan pada Tabel 6.

**Tabel 6.** Hasil Uji Akurasi Identifikasi Frekuensi

<b>Rentang Frekuensi</b>	<b>Akurasi</b>	<b>Mean Error</b>	<b>Max Error</b>
80-200 Hz	95.7%	6.0 Hz	6 Hz
200-400 Hz	96.3%	7.5 Hz	9 Hz
400-600 Hz	97.3%	5.5 Hz	9 Hz
600-800 Hz	98.5%	3.0 Hz	4 Hz
800-1100 Hz	98.8%	3.5 Hz	4 Hz

Hasil menunjukkan performa terbaik pada rentang frekuensi tinggi (800-1100 Hz) dengan akurasi 98.8% dan *mean error* 3.5 Hz. Performa terendah terjadi pada rentang frekuensi rendah (80-200 Hz) dengan akurasi 95.7%, yang dapat disebabkan oleh sensitivitas *noise* pada frekuensi rendah dan resolusi *FFT* yang terbatas.



**Gambar 3.** Grafik Spektrum dan Hasil Uji pada Frekuensi 200 Hz

Berdasarkan Gambar 3 hasil deteksi pada frekuensi 200 Hz, sistem mengidentifikasi *peak dominan* pada 193.6 Hz (G3) dengan amplitudo sekitar 800. Hasil ini menunjukkan selisih 6.4 Hz dari target 200 Hz, yang konsisten dengan akurasi 96.3% pada rentang 200-400 Hz dan *mean error* 7.5 Hz, serta berhasil mengklasifikasikan tipe suara sebagai Baritone.

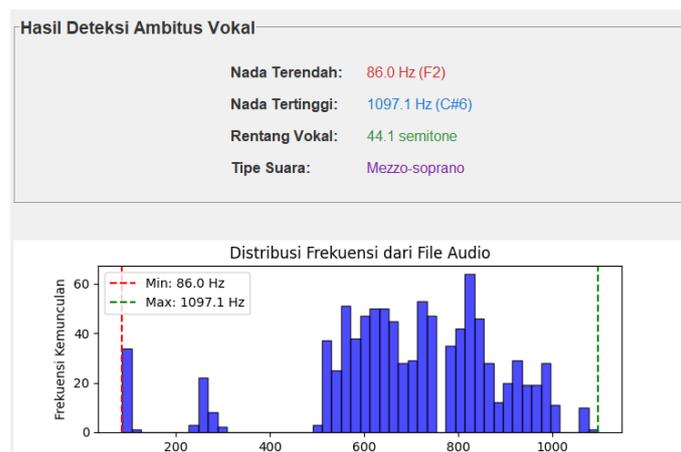
### 3.4. Evaluasi Klasifikasi Tipe Suara

Evaluasi klasifikasi tipe suara dilakukan menggunakan *dataset VocalSet* dengan 30 sampel audio dari 6 kategori tipe suara [8]. Hasil evaluasi menunjukkan akurasi keseluruhan 81.2% dengan variasi performa per kategori seperti yang ditunjukkan pada Tabel 7.

**Tabel 7.** Akurasi Klasifikasi Tipe Suara

Tipe Suara	Jumlah Sampel	Akurasi	Precision	Recall
Bass	5	88%	0.88	0.88
Baritone	5	75%	0.75	0.75
Tenor	5	89%	0.89	0.89
Alto	5	67%	0.67	0.67
Mezzo-soprano	5	90%	0.90	0.90
Soprano	5	72%	0.72	0.72

Berdasarkan Tabel 7, performa terbaik dicapai pada kategori *Mezzo-soprano* (90%) dan *Tenor* (89%), sementara performa terendah pada kategori *Alto* (67%). Perbedaan ini disebabkan oleh *overlap* karakteristik frekuensi antara beberapa kategori vokal, terutama pada rentang tengah yang sering menjadi *ambiguous boundary* antar kelas.



**Gambar 4.** Grafik Spektrum dan Hasil Uji pada Tipe Suara Mezzo-soprano

Berdasarkan Gambar 4 hasil deteksi pada sampel Mezzo-soprano, sistem berhasil mengidentifikasi rentang vokal yang luas dengan nada terendah 86.0 Hz (F2) dan nada tertinggi 1097.1 Hz (C#6), menghasilkan rentang vokal 44.1 semitone. Distribusi frekuensi menunjukkan pola spektral yang kompleks dengan *multiple peaks* pada rentang 86-1097 Hz, termasuk *peak* dominan di sekitar 600-700 Hz dan aktivitas frekuensi yang tersebar hingga rentang tinggi. Sistem berhasil mengklasifikasikan sampel ini sebagai Mezzo-soprano dengan benar, yang konsisten dengan performa kategori ini yang mencapai akurasi tertinggi 90% dalam evaluasi, menunjukkan kemampuan sistem dalam mengidentifikasi karakteristik spektral yang khas untuk tipe suara wanita menengah.

### 3.5. Perbandingan dengan Metode Lain

Sistem yang dikembangkan dibandingkan dengan metode identifikasi vokal lainnya untuk mengevaluasi efektivitas pendekatan *FFT*. Perbandingan mencakup aspek akurasi, kompleksitas komputasi, dan kemudahan implementasi seperti yang ditunjukkan pada Tabel 8.

**Tabel 7.** Perbandingan Metode Identifikasi Ambitus Vokal

Metode	Akurasi	Kompleksitas	Waktu Pemrosesan	Kebutuhan Data
<i>Deep Learning</i> [5]	92-95%	$O(n^3)$	50-100 ms	Besar (>10GB)
<i>Spectral Analysis</i> [4]	85-90%	$O(n^2)$	20-40 ms	Sedang (1-5GB)
<b><i>FFT</i> (Penelitian Ini)</b>	<b>81.2%</b>	<b><math>O(n \log n)</math></b>	<b>5 ms</b>	<b>Minimal (&lt;100MB)</b>

Berdasarkan Tabel 8, sistem *FFT* yang dikembangkan menunjukkan *trade-off* yang optimal antara akurasi (81.2%) dan efisiensi komputasi (5 ms *processing time*). Meskipun akurasi lebih rendah dibandingkan *deep learning methods*, sistem ini menawarkan keunggulan dalam hal kecepatan pemrosesan, kompleksitas komputasi yang rendah, dan kebutuhan data *training* yang minimal, menjadikannya solusi praktis untuk aplikasi *real-time*.

### 3.6. Limitasi dan Tantangan

Beberapa limitasi sistem yang teridentifikasi meliputi: sensitivitas terhadap *noise* lingkungan yang mempengaruhi akurasi identifikasi pada frekuensi rendah, kesulitan dalam menangani teknik vokal vibrato yang menyebabkan fluktuasi frekuensi, dan *overlapping* karakteristik frekuensi antar kategori vokal yang menyebabkan ambiguitas klasifikasi. Tantangan tambahan meliputi variasi individual dalam karakteristik vokal dan pengaruh faktor eksternal seperti kualitas mikrofon dan kondisi akustik ruangan.

#### 4. Kesimpulan

Penelitian ini berhasil mengembangkan sistem identifikasi dan klasifikasi ambitus vokal berbasis algoritma *FFT* dengan mencapai akurasi identifikasi frekuensi 95.7-98.8% dan akurasi klasifikasi jenis suara 81.2%. Sistem mampu melakukan pemrosesan *real-time* dengan *latency* rendah (5 ms) dan penggunaan sumber daya komputasi yang minimal, sehingga memberikan solusi praktis untuk analisis karakteristik vokal dengan keseimbangan optimal antara akurasi dan efisiensi komputasi.

#### Daftar Pustaka

- [1] R. Monir, D. Kostrzewa, dan D. Mrozek, "Singing Voice Detection: A Survey," *Entropy*, vol. 24, no. 1, 2022, doi: 10.3390/e24010114. <https://www.mdpi.com/1099-4300/24/1/114>
- [2] R. T. Sataloff, "Professional Singers: The Science and Art of Clinical Care," *Otolaryngology- Head and Neck Surgery*, vol. 112, no. 5, hlm. 51, 1995, doi: 10.1016/S0194-5998(05)80097-X. <https://aao-hnsfjournals.onlinelibrary.wiley.com/doi/10.1016/S0194-5998%2805%2980097-X>
- [3] Alejandra Arevalo, "TikTok or YouTube Singer, Influencers Are Becoming Artists," How Music Charts. Diakses: 28 Juni 2025. [Daring]. Tersedia pada: <https://hmc.chartmetric.com/tiktok-youtube-singer/>
- [4] J. Sun, "Research on vocal sounding based on spectrum image analysis," *EURASIP J Image Video Process*, vol. 2019, no. 1, 2019, doi: 10.1186/s13640-018-0397-0. <https://jivp-urasipjournals.springeropen.com/articles/10.1186/s13640-018-0397-0>
- [5] X. Zhang, Y. Yu, Y. Gao, X. Chen, dan W. Li, "Research on singing voice detection based on a long-term recurrent convolutional network with vocal separation and temporal smoothing," *Electronics (Switzerland)*, vol. 9, no. 9, hlm. 1–23, 2020, doi: 10.3390/electronics9091458. <https://www.mdpi.com/2079-9292/9/9/1458>
- [6] M. Müller, "Fundamentals of Music Processing," *Fundamentals of Music Processing*, 2015, doi: 10.1007/978-3-319-21945-5. <https://link.springer.com/10.1007/978-3-319-21945-5>
- [7] B. McFee dkk., "librosa: Audio and Music Signal Analysis in Python," *Proceedings of the 14th Python in Science Conference*, no. Scipy, hlm. 18–24, 2015, doi: 10.25080/majora-7b98e3ed-003. <https://doi.curvenote.com/10.25080/Majora-7b98e3ed-003>
- [8] J. Wilkins, P. Seetharaman, A. Wahl, dan B. Pardo, "Vocalset: A singing voice dataset," *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, hlm. 468–474, 2018. [https://ismir2018.ircam.fr/doc/pdfs/114\\_Paper.pdf](https://ismir2018.ircam.fr/doc/pdfs/114_Paper.pdf)
- [9] D. T. Kusuma, "Fast Fourier Transform (FFT) Dalam Transformasi Sinyal Frekuensi Suara Sebagai Upaya Perolehan Average Energy (AE) Musik," *Petir*, vol. 14, no. 1, hlm. 28–35, 2020, doi: 10.33322/petir.v14i1.1022. <http://jurnal.itpln.ac.id/petir/article/view/1022>
- [10] B. Lehner, R. Sonnleitner, dan G. Widmer, "Singing Voice Detection," *2018 International Society for Music Information Retrieval Society Conference*, hlm. 1–6. <https://encyclopedia.pub/entry/18717>