

ARTICLE

Algoritma Fourier Transformation Untuk Mendeteksi Suara Senar Alat Musik Akustik Berbasis Web

Fourier Transformation Algorithm for Detecting Acoustic Musical Instrument String Sounds Web-Based

Timothy James Wijaya* dan Fransiskus Adikara

Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia, Jakarta, Indonesia

*Penulis Korespondensi: fadikara@bundamulia.ac.id

(Disubmit 12-08-24; Diterima 11-09-24; Dipublikasikan online pada 05-02-25)

Abstrak

Alat musik akustik seperti gitar dan biola memegang peranan penting dalam dunia musik. Kualitas suara yang dihasilkan oleh alat musik ini sangat bergantung pada kondisi senarnya. Penyetelan yang tidak tepat, baik karena ketegangan yang tidak tepat atau kerusakan fisik, dapat menghambat performa musisi dan menghasilkan suara yang tidak diinginkan. Untuk mengatasi permasalahan tersebut, penelitian ini mengembangkan sebuah aplikasi berbasis website untuk menganalisis suara real-time, yang dapat mendeteksi suara-suara yang tidak sesuai dari senar gitar. Perangkat lunak ini memberikan umpan balik langsung kepada pemain dan mengidentifikasi masalah intonasi pada senar gitar saat senar dipetik. Penelitian ini juga mengevaluasi keakuratan aplikasi dalam mendeteksi kesalahan pada senar gitar menggunakan metode pengujian berdasarkan kasus permainan dan kondisi senar yang berbeda. Hasil pengujian yang diharapkan dari penelitian ini dapat memberikan hasil yang akurat dalam mendeteksi masalah suara senar gitar, yang dapat membantu para pecinta musik menjaga kualitas suara instrumennya dan meningkatkan kinerja musiknya.

Kata kunci: Pemrograman Website; Fast Fourier Transformation; Analisis Audio Real-Time; Deteksi Suara Senar; Alat Musik Akustik

Abstract

Acoustic musical instruments such as guitars and violins play an important role in the world of music. The quality of the sound produced by this musical instrument depends on the condition of the strings. Improper tuning, whether due to improper tension or physical damage, can hinder a musician's performance and produce undesirable sounds. To overcome this problem, this research developed a website-based application with real-time sound analysis, which can detect inappropriate sounds from the guitar strings. This software provides direct feedback to the player and identifies intonation problems on the guitar strings as the strings are plucked. This research also evaluates the accuracy of the application in detecting errors in guitar strings using test methods based on different playing cases and string conditions. It is hoped that the test results from this research will provide accurate results in detecting acoustic instrument string sound problems, specially guitar, which can help music lovers maintain the sound quality of their instruments and improve their musical performance.

KeyWords: Website Programming; Fast Fourier Transformation; Audio Real-Time Analysis; String Sound Detection; Acoustic Instrument

This is an Open Access article - copyright on authors, distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY SA) (<http://creativecommons.org/licenses/by-sa/4.0/>)

How to Cite: T. J. Wijaya *et al.*, "Algoritma Fourier Transformation Untuk Mendeteksi Suara Senar Alat Musik Akustik Berbasis Web", *JIKO (JURNAL INFORMATIKA DAN KOMPUTER)*, Volume: 9, No.1, Pages 79–92, Februari 2025, doi: 10.26798/jiko.v9i1.1411.

1. Pendahuluan

Alat musik akustik seperti gitar, biola atau piano telah menjadi bagian penting dalam dunia musik dan seni. Integritas suara yang dihasilkan alat musik ini sangat bergantung pada kondisi senarnya. Pengaturan senar yang tidak tepat, baik karena ketegangan yang tidak tepat atau kerusakan fisik, dapat menimbulkan suara yang tidak diinginkan dan menghambat performa pemain musik. Untuk mengatasi permasalahan tersebut diperlukan suatu solusi yang dapat mendeteksi dengan cepat dan akurat apakah suara yang dihasilkan oleh senar suatu instrumen akustik memenuhi standar yang diharapkan atau sebaliknya [1].

Dalam konteks ini, algoritma untuk menganalisis *Audio Real-Time* menjadi sangat relevan. Teknologi analisis audio telah berkembang pesat dalam beberapa tahun terakhir dan kini memungkinkan penggunaan perangkat lunak yang dapat secara otomatis mendeteksi masalah pada senar instrumen akustik, khususnya alat musik gitar. Salah satu teknologi Analisis Audio adalah memanfaatkan perangkat lunak yang terhubung dengan mikrofon sebagai media *input* untuk mendapatkan *sample* suara lalu menerjemahkannya menjadi bentuk nilai yang diinginkan. Dengan menggunakan teknologi ini, para pecinta musik dapat memastikan suara instrumennya tetap pada level optimal sehingga meningkatkan kualitas penampilan musiknya.

Lebih lanjut, pengembangan perangkat lunak yang mempunyai kemampuan analisis audio tersebut berpotensi untuk mendukung pendidikan musik, sehingga guru atau instruktur dapat menggunakan website tersebut sebagai alat bantu dalam mendidik siswa tentang pentingnya intonasi dan nada pada instrumen akustik. Selain itu pengguna alat musik juga dapat menggunakan aplikasi ini untuk memaksimalkan kualitas suara dari senar gitar dengan intonasi dan nada yang diinginkan. Perangkat lunak yang dikembangkan diharapkan mempunyai akurasi yang tinggi dalam mendeteksi nada dari senar instrumen akustik dalam bentuk nilai frekuensi dan notasi dari suara menggunakan *algoritma Fourier Transform* untuk analisis *Audio Real-Time*.

Manfaat yang didapatkan dari penelitian ini adalah *software* yang dikembangkan dapat membantu para pecinta musik dalam menjaga kualitas suara alat musik akustik mereka dengan mendeteksi kesalahan pada senar secara *real-time*. Hal ini memungkinkan untuk mereka dapat menghasilkan performa musik yang lebih baik dengan kondisi alat musik yang optimal.

2. Metode

Pada penelitian ini, metodologi penelitian yang dilaksanakan adalah:

1. Studi Pustaka untuk mendapatkan tinjauan pustaka dari *Algoritma Fourier Transformation* untuk melakukan analisis *Audio Real-Time*.
2. Studi Pustaka untuk mendapatkan tinjauan pustaka dari penelitian terkait *software analisis Audio Real-Time* yang pernah dikembangkan sebelumnya.
3. Menggunakan metode *Rapid Application Development* untuk mengembangkan perangkat lunak ini.
4. Menguji sistem yang dibangun apakah sudah memenuhi kebutuhan fungsional dan non-fungsional yang sudah ditetapkan.

Berdasarkan metodologi di atas, bagian makalah ini berikutnya akan dibagi menjadi pembahasan mengenai Tinjauan Pustaka (langkah 1 dan 2), dilanjutkan dengan proses pembangunan *software* dan hasilnya (langkah 3 dan 4), diakhiri dengan kesimpulan.

2.1 Algoritma Fourier Transformation

Software ini menggunakan metode *Fourier Transform*, sehingga saat suara senar gitar terdeteksi oleh mikrofon yang terhubung dengan perangkat pengguna, suara akan dibaca oleh sistem dan suara akan ditranslasikan menjadi bentuk nilai notasi dan frekuensi. Notasi dan frekuensi yang akan ditampilkan dengan akurasi yang mendekati nilai frekuensi dari masing-masing suara notasi yang terdeteksi agar pengguna dapat mendapatkan hasil *tuning* gitar yang diharapkan.

Algoritma ini sesuai dengan gambaran *Guitar Tuner* yang akan dibuat di mana suara notasi yang terdeteksi pada *microphone* akan dikonversikan menjadi nilai yang bernilai satuan frekuensi. Ide di balik dekomposisi *Fourier* adalah untuk memecah bentuk gelombang yang rumit menjadi beberapa bentuk gelombang sederhana (*sinusoidal*), yang masing-masing hanya mewakili satu frekuensi. Ini melibatkan penghitungan sekumpulan besaran yang terlihat seperti pada persamaan (1).

$$\sum \phi(t)\sin(2\pi ft) \tag{1}$$

- di mana :
- $\phi(t)$: Mewakili amplitudo suara sebagai fungsi waktu, dan *sinusoidal* adalah gelombang dengan frekuensi tertentu.
- f : Frekuensi

Secara umum, persamaan (1) dapat digunakan untuk menghitung nilai frekuensi dari suara gelombang apa pun. Nilai frekuensi yang dihasilkan akan menjadi frekuensi yang paling dominan dalam suara gelombang tersebut.

Kita dapat asumsi-kan jika ϕ naik dan turun dengan cara yang sama seperti *sinusoidal*, maka penjumlahannya adalah sekumpulan nilai positif. Di sisi lain, jika ϕ berbeda, maka syarat penjumlahannya akan menghasilkan nilai acak yang meniadakan nol (tidak menumpuk nilai semua positif ataupun semua negatif). Berikut ini adalah contoh dari persamaan (1): Suara gelombang sinus murni dengan frekuensi 440 Hz:
 $\phi(t) = \sin(2\pi * 440 * t)$

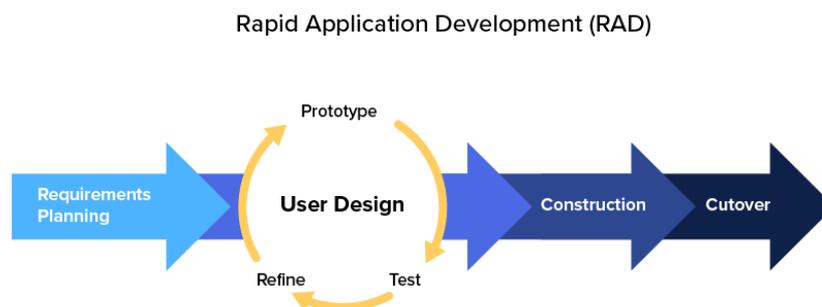
Dengan rumus ini, kita dapat menghitung nilai frekuensi dari suara gelombang sinus murni ini
 $\sum \phi(t)\sin(2\pi ft) = \sum \sin(2\pi * 440 * t)\sin(2\pi ft)$.

Nilai $\sum \phi(t)\sin(2\pi ft)$ hanya akan bernilai positif jika $f = 440$ dan nilai $\sum \phi(t)\sin(2\pi ft)$ akan bernilai nol jika f berbeda dari 440. Pengaplikasian dari rumus ini salah satunya untuk menentukan *pitch*(nada) dari suara gitar[1].

2.2 Rapid Application Development

Software ini menggunakan metode *Fourier Transform*, sehingga saat suara senar gitar terdeteksi oleh mikrofon yang terhubung dengan perangkat pengguna suara akan dibaca oleh sistem dan suara akan ditranslasi menjadi bentuk nilai notasi dan frekuensi. *Rapid Application Development* merupakan sebuah proses yang digunakan untuk merancang dan mengembangkan penelitian ini. Dalam metode RAD ini terdapat empat(4) tahapan, yaitu membuat perencanaan kebutuhan(*Requirements Planning*), desain pengguna (*User Design*), pengembangan (*Construction*), dan implementasi (*Cutover*)[4].

Gambar 1 berikut adalah gambar dari langkah-langkah dalam metode RAD:



Gambar 1. RAD

1. **Perencanaan kebutuhan(Requirement Planning)**

Pada tahap ini penulis menentukan tujuan dari penelitian ditentukan dan rencana kerja disusun. Analisis keperluan dan tujuan penelitian yang mendukung proses perancangan perangkat lunak, pengguna yang dituju adalah orang-orang yang memiliki pengalaman tentang musik maupun pengguna masih baru ataupun sudah pakar dan juga audio/gambar yang akan digunakan dalam merancang *software*.

2. **desain pengguna(User Design)**

Tahap ini dimana penulis membuat rancangan yang akan diusulkan untuk memenuhi kebutuhan dari seorang pengguna, berjalan sesuai dengan perencanaan yang telah ditentukan, dan dapat menjawab atau mengatasi masalah yang dibawakan dalam penelitian ini. Pada penelitian ini, desain sistem akan digambarkan menggunakan *Unified Modeling Language (UML)*.

3. **Pengembangan (Construction)**

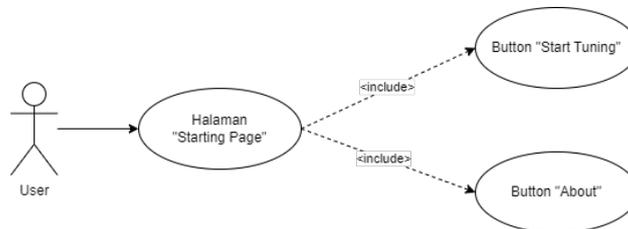
Pada tahap ini, penulis memulai membuat *software* dengan bahasa pemrograman HTML sesuai dengan desain sistem yang telah direncanakan dan dirancang. *Software* ini akan digunakan oleh penulis untuk melakukan penelitian yaitu mengimplementasikan *algoritma fourier transformation* ke dalam perangkat lunak yang dirancang dan mengetahui akurasi dari *algoritma fourier transformation* dalam menghasilkan nilai frekuensi dan nama note yang tepat dan akurat.

4. **Implementasi (Cutover)**

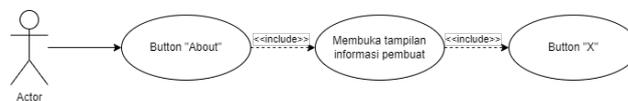
Pada tahap ini penulis mengimplementasikan perangkat lunak mulai dari perencanaan, *user requirement* dan juga *design* yang sudah dirancang ke dalam *software* yang akan dibuat untuk penelitian dan mempublikasikan demo *software* melalui *platform Github*.

2.3 Proses Perancangan Desain

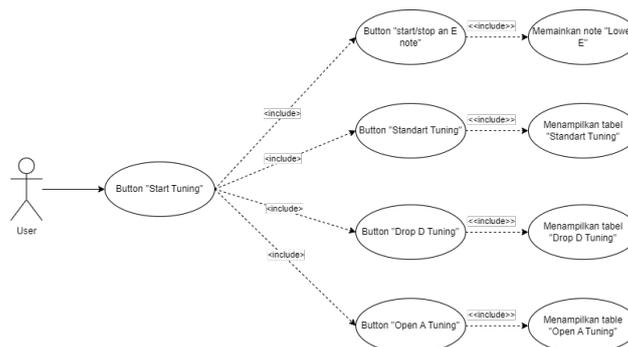
Use Case Diagram *Use Case Diagram* adalah sebuah diagram yang menggambarkan sebuah interaksi antara sistem dan juga pengguna atau aktor, terlihat pada Gambar 2, 3 dan 4 (baik itu individu atau sebuah sistem) untuk mengidentifikasi fungsionalitas sistem [8].



Gambar 2. Use Case Diagram Starting Page

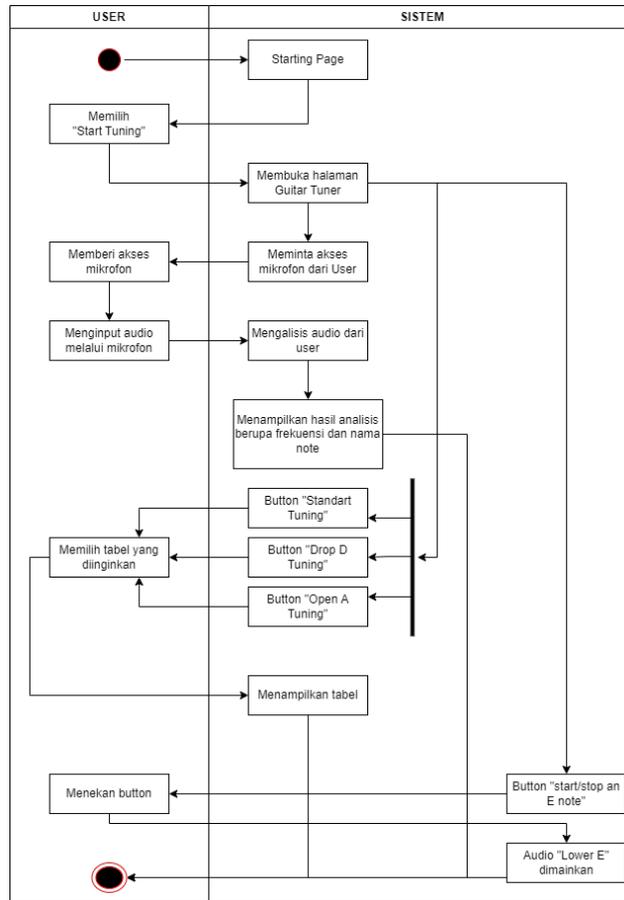


Gambar 3. Use Case Diagram Button About

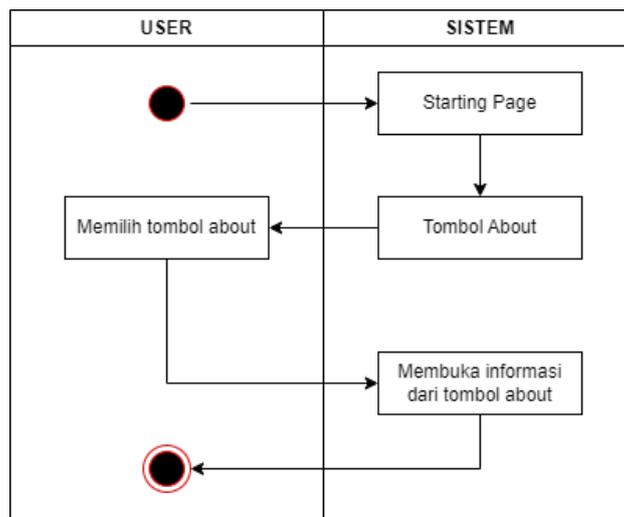


Gambar 4. Use Case Diagram Button Start Tuning

Activity Diagram Activity Diagram berfungsi untuk menggambarkan alur kerja dalam sistem, menunjukkan aktivitas dan juga transisi antar masing-masing aktivitas terlihat pada Gambar 5 dan 6. Kegiatan alur kerja menunjukkan perilaku dari sistem terhadap aktivitas yang ada [8].

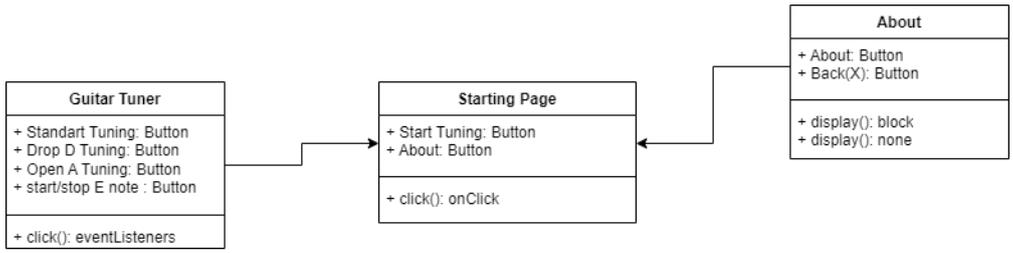


Gambar 5. Activity Diagram Menu Gitar Tuner



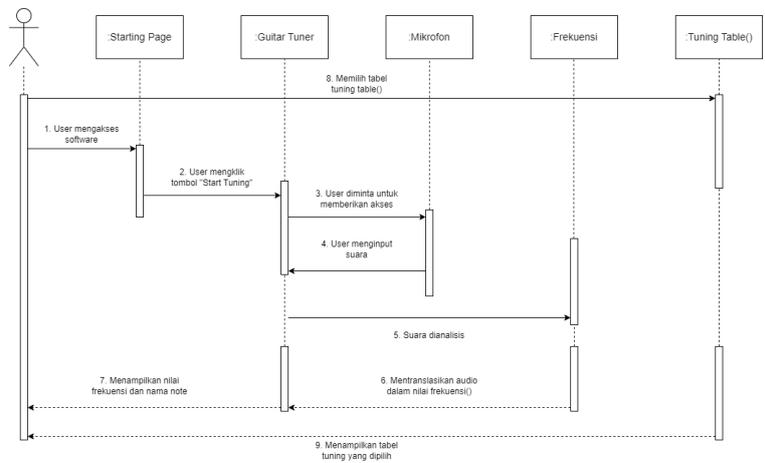
Gambar 6. Activity Diagram Menu About

Class Diagram Class Diagram menunjukkan struktur statis dari sebuah sistem, termasuk kelas, atribut, operasi, dan hubungan antar kelas terlihat pada Gambar 7. Class Diagram sering digunakan karena mampu membantu untuk menunjukkan struktur dari kelas-kelas pada sistem yang dirancang [8].

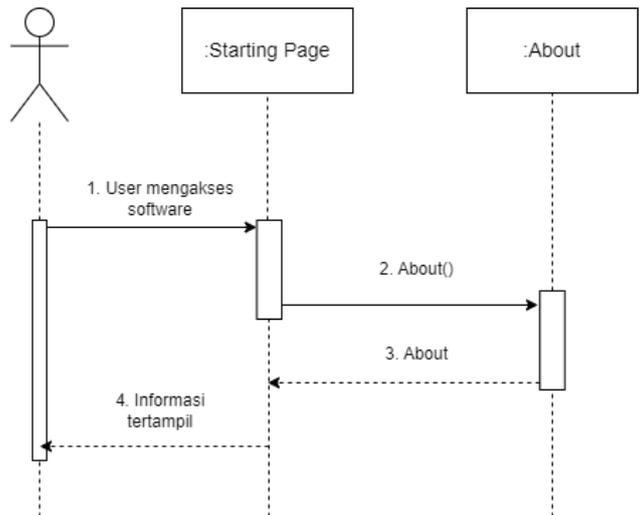


Gambar 7. Class Diagram Software

Sequence Diagram *Sequence Diagram* adalah diagram yang menunjukkan urutan interaksi yang dimulai dari aktor melakukan interaksi antara objek yang telah disusun berdasarkan urutan waktu [8]. Gambar 8 dan 8, *sequence diagram* untuk *Tuner* dan *About* dari *software*.



Gambar 8. Sequence Diagram Guitar Tuner



Gambar 9. Sequence Diagram About

2.4 Proses Perancangan Desain

Whitebox Testing adalah pengujian *software* pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan [6]. Dalam pengujian, uji dirancang dari perspektif pengembang dikarenakan struktur internal dikenal dengan menguji segala bagian kode yang mampu untuk diuji dengan tujuan untuk menentukan kesalahan logis dari kode sumber perangkat lunak [8].

3. Hasil

3.1 Pengujian Keberhasilan Algoritma Fourier Transform

Berikut dibawah ini adalah hasil pengujian keberhasilan dari algoritma *Fourier Transform* yang diimplementasikan ke dalam *software* yang dikembangkan. Percobaan dilakukan sampai semua nada yang diminta sudah sesuai dengan nilai frekuensi yang ada, terlihat pada Tabel 1, 2, 3, 4, 5, 6.

Tabel 1. Tabel Standart Tuning

No	Frekuensi	Nama Nada	Frekuensi Tertampil	Hasil
Senar 6	E (82.41 Hz)	E	82,41143587362365	Sesuai
Senar 5	A (110.00 Hz)	A	110,00606904349094	Sesuai
Senar 4	D (146.83 Hz)	D	146,84048515983218	Sesuai
Senar 3	G (196.00 Hz)	G	196,00853179700746	Sesuai
Senar 2	B (246.94 Hz)	B	246,95527517003828	Sesuai
Senar 1	E (329.63 Hz)	E	329,64574349449464	Sesuai

Tabel 2. Tabel Drop D Tuning

No	Frekuensi	Nama Nada	Frekuensi Tertampil	Hasil
Senar 6	D (73.42 Hz)	D	73,42024257991605	Sesuai
Senar 5	A (110.00 Hz)	A	110,00606904349094	Sesuai
Senar 4	D (146.83 Hz)	D	146,84048515983218	Sesuai
Senar 3	G (196.00 Hz)	G	196,00853179700746	Sesuai
Senar 2	B (246.94 Hz)	B	246,95527517003828	Sesuai
Senar 1	D (293.66 Hz)	D	293,68097031966425	Sesuai

Tabel 3. Tabel Open A Tuning

No	Frekuensi	Nama Nada	Frekuensi Tertampil	Hasil
Senar 6	E (82.41 Hz)	E	82,41143587362365	Sesuai
Senar 5	A (110.00 Hz)	A	110,00606904349094	Sesuai
Senar 4	E (164.81 Hz)	E	164,822871747473	Sesuai
Senar 3	A (220.00 Hz)	A	220,0121380869188	Sesuai
Senar 2	C# (277.18 Hz)	C#	277,197924008166	Sesuai
Senar 1	E (329.63 Hz)	E	329,64574349449464	Sesuai

Tabel 4. Tabel Standart Tuning (Senar String)

No	Frekuensi	Nama Nada	Frekuensi Tertampil	Hasil
Senar 6	E (82.41 Hz)	E	82.41143587362365	Sesuai
Senar 5	A (110.00 Hz)	A	110.00606904349094	Sesuai
Senar 4	E (164.81 Hz)	E	146.84048515983218	Sesuai
Senar 3	A (220.00 Hz)	A	196.00853179700746	Sesuai
Senar 2	C# (277.18 Hz)	C#	246.95527517003828	Sesuai
Senar 1	E (329.63 Hz)	E	329.64574349449464	Sesuai

Tabel 5. Tabel Drop D Tuning (Senar String)

No	Frekuensi	Nama Nada	Frekuensi Tertampil	Hasil
Senar 6	E (82.41 Hz)	E	73.42024257991605	Sesuai
Senar 5	A (110.00 Hz)	A	110.00606904349094	Sesuai
Senar 4	E (164.81 Hz)	E	146.84048515983218	Sesuai
Senar 3	A (220.00 Hz)	A	196.00853179700746	Sesuai
Senar 2	C# (277.18 Hz)	C#	246.95527517003828	Sesuai
Senar 1	E (329.63 Hz)	E	293.68097031966425	Sesuai

Tabel 6. Tabel Open A Tuning (Senar String)

No	Frekuensi	Nama Nada	Frekuensi Tertampil	Hasil
Senar 6	E (82.41 Hz)	E	82.41143587362365	Sesuai
Senar 5	A (110.00 Hz)	A	110.00606904349094	Sesuai
Senar 4	E (164.81 Hz)	E	164.8228717472473	Sesuai
Senar 3	A (220.00 Hz)	A	220.0121380698188	Sesuai
Senar 2	C# (277.18 Hz)	C#	277.197924008166	Sesuai
Senar 1	E (329.63 Hz)	E	329.64574349449464	Sesuai

3.2 Pengujian Keberhasilan Tombol

Tabel 7. Tabel Pengujian Keberhasilan Tombol Tampilan Menu Utama

No	Kegiatan	Input	Output	Hasil
1	Melihat informasi about	Menekan tombol 'About'	Panel about tertampil	Sesuai
2	Memasuki halaman Gitar Tuner	Menekan tombol 'Start Tuning'	Berpindah ke halaman Gitar Tuner	Sesuai

Tabel 8. Tabel Pengujian Keberhasilan Tombol Tampilan Halaman Tuner Gitar

No	Kegiatan	Input	Output	Hasil
1	Mendengarkan suara note E2	Menekan tombol 'start/stop an E note'	Suara note E2 dimainkan	Sesuai
2	Melihat tabel Standart Tuning	Menekan tombol 'Standart Tuning'	Menampilkan tabel Standart Tuning	Sesuai
3	Melihat tabel Drop D Tuning	Menekan tombol 'Drop D Tuning'	Menampilkan tabel Drop D Tuning	Sesuai
4	Melihat tabel Open A Tuning	Menekan tombol 'Open A Tuning'	Menampilkan tabel Open A Tuning	Sesuai

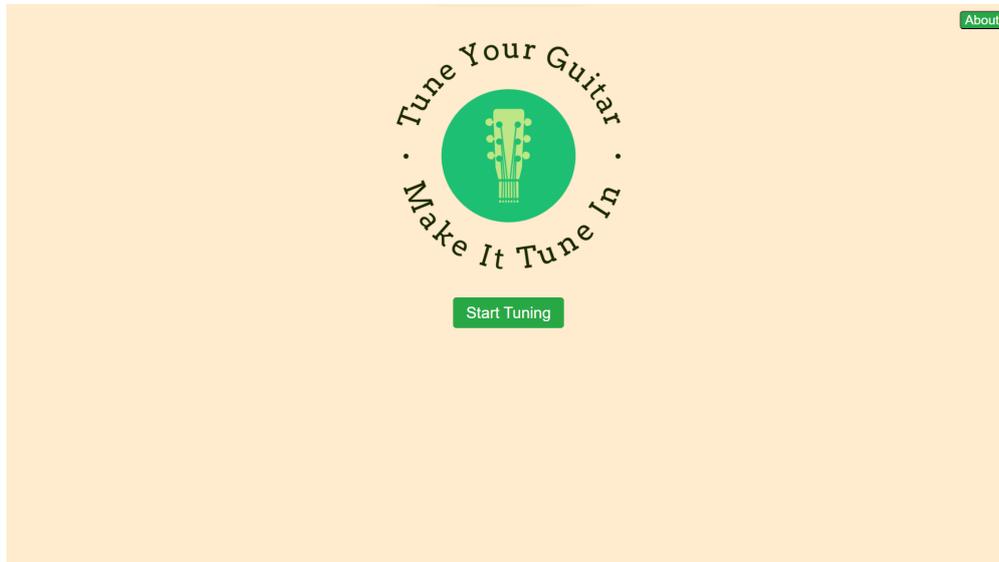
3.3 Diskusi Hasil Pengujian Keberhasilan Algoritma Fourier Transform

Algoritma Fourier Transform (FT) telah banyak digunakan dalam berbagai aplikasi, termasuk pemrosesan sinyal dan analisis frekuensi, namun tidak lepas dari tantangan dalam lingkungan tertentu. Salah satu kendala utama adalah keberadaan kebisingan atau *noise*, yang dapat mempengaruhi akurasi transformasi frekuensi. *Noise* acak yang tumpang tindih dengan sinyal utama dapat mengaburkan informasi *spektral* penting, sehingga sulit memisahkan komponen sinyal yang relevan dari gangguan tersebut, walaupun dalam penelitian ini, sebagian besar dilakukan di ruang tertutup sehingga hasilnya menjadi lebih akurat. Dalam lingkungan akustik yang kompleks, seperti ruang yang memiliki pantulan atau reverberasi tinggi, FT juga dapat menghadapi kesulitan dalam mendeteksi sinyal asli karena interferensi dari sinyal-sinyal pantulan tersebut, oleh sebab itu pada

penelitian kali ini, penulis memastikan tidak ada gema yang terjadi pada ruang pengujian dan menggunakan mikrofon yang mempunyai daya tangkap baik. Oleh karena itu, untuk aplikasi di lingkungan dengan *noise* atau kompleksitas akustik tinggi, pengembangan algoritma yang lebih adaptif terhadap dinamika sinyal sangat diperlukan.

3.4 Hasil Perancangan

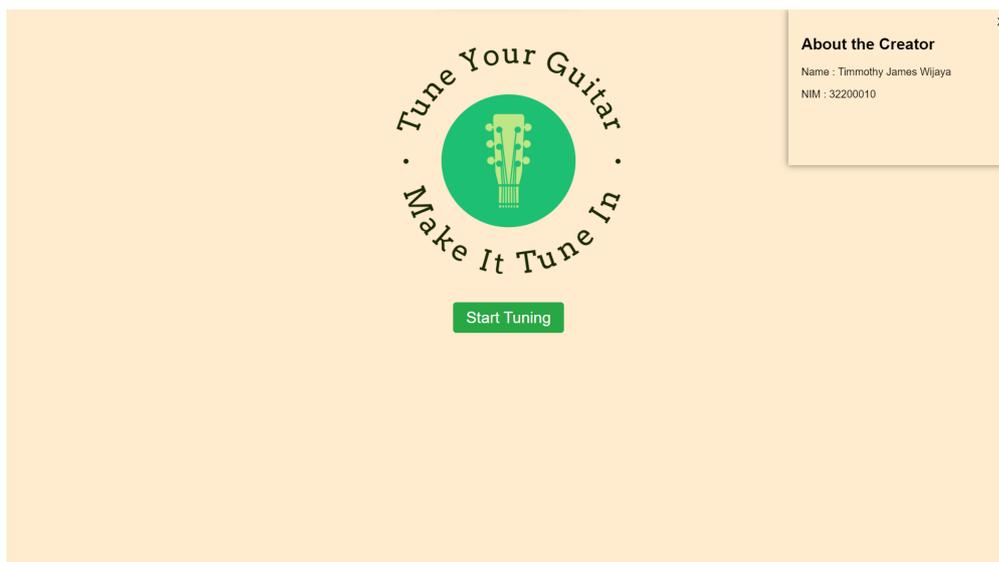
3.4.1 Halaman Utama



Gambar 10. Tampilan Menu Utama

Gambar 10 di atas merupakan tampilan untuk menu utama yang di mana tampilan ini akan muncul pertama kali saat *software* diakses oleh pengguna. Dalam tampilan tersebut terdapat logo 'Tune Your Guitar', tombol 'Start Tuning', dan tombol 'About'.

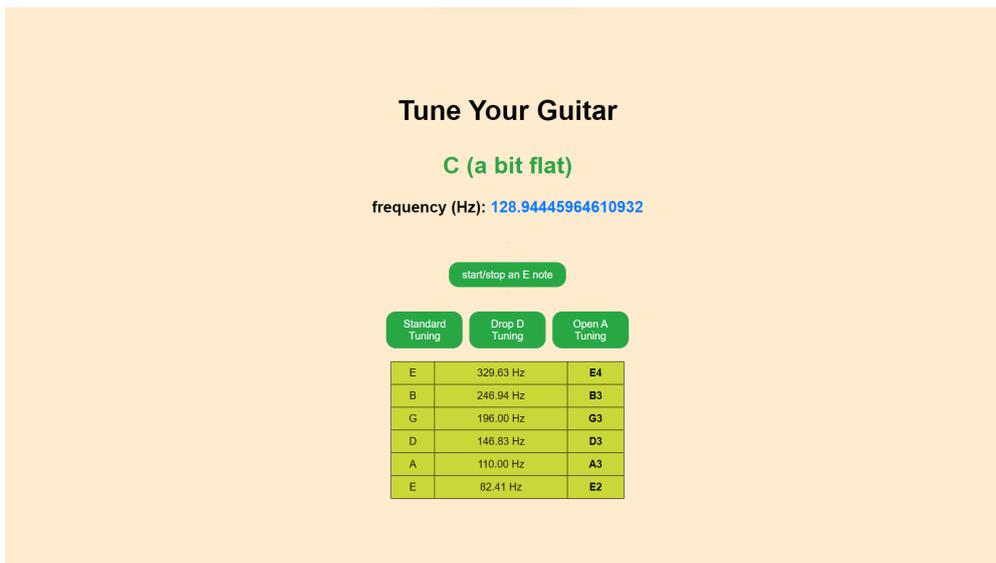
3.4.2 Tampilan About



Gambar 11. Tampilan About

Gambar 11 di atas merupakan tampilan dari panel *about* yang muncul saat pengguna mengklik tombol 'About' yang di mana isinya terdapat informasi singkat pembuat *software*.

3.4.3 Tampilan Halaman Tuner Gitar



Gambar 12. Tampilan Halaman Tuner Gitar

Gambar 12 di atas merupakan tampilan untuk halaman *tuner* gitar di mana pengguna dapat memulai menggunakan *software* 'Tune Your Guitar'. Pengguna akan diminta untuk memberikan akses mikrofon pada perangkat yang digunakan oleh pengguna agar *software* dapat mendeteksi *input* suara yang diberikan oleh pengguna. Suara yang telah terdeteksi oleh sistem pada *software* akan ditranslasikan ke dalam bentuk nama nada dan juga nilai frekuensi yang akan tampil. Terdapat juga *button* 'start/stop an E notasi' yang di mana pengguna akan mendengarkan suara notasi E untuk mendapatkan visualisasi suara. Terdapat tiga buah *button* juga yang berfungsi untuk menampilkan tabel yang berisikan nama notasi, nilai frekuensi, dan juga *scientific pitch* notasi yang berfungsi untuk membantu pengguna dalam proses *tuning*. Proses translasi suara yang terdeteksi menjadi sebuah nilai frekuensi menggunakan *algoritma Fourier Transform*.

3.5 Usability Testing

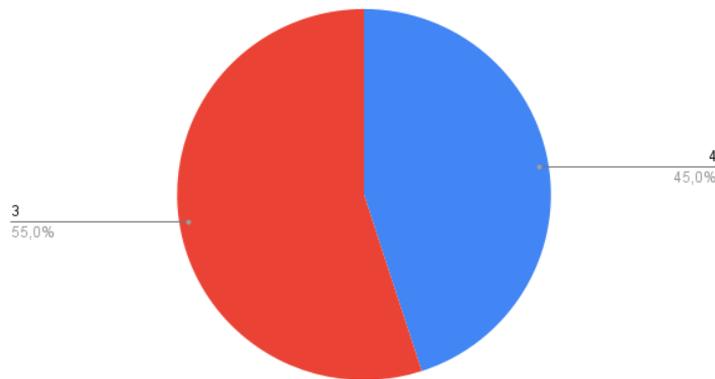
Teknik pengumpulan data dilakukan menggunakan kuesioner yang di mana dibagikan kepada 20 responden yang mengenal perangkat lunak *guitar tuner*. Kuesioner ini bertujuan untuk mengukur seberapa akurat perangkat lunak mendeteksi suara nada senar menjadi nilai frekuensi dan nama notasi sesuai dengan penilaian yang dilakukan oleh responden. Perhitungan untuk kuesioner dilakukan dengan perhitungan *skala likert* yang di mana setiap jawaban memiliki nilai tertentu yaitu 4 untuk nilai "Sangat Setuju", 3 untuk nilai "Setuju", 2 untuk nilai "Tidak Setuju" dan 1 untuk nilai "Sangat Tidak Setuju".

Tabel 9. Akumulasi Nilai Evaluasi Software - 1

Nilai	Jumlah Responden	Akumulasi Nilai Pertanyaan Evaluasi Software - 1
1	0	0
2	0	0
3	10	30
4	10	40
Total	20	70

$$Y = 20 * 4 = 80, \text{ Indeks Evaluasi Software - 1(\%)} = 7080 * 100 = 87,5\%$$

Suara yang Saya input terdeteksi secara akurat



Gambar 13. Diagram Lingkaran Evaluasi Software-1

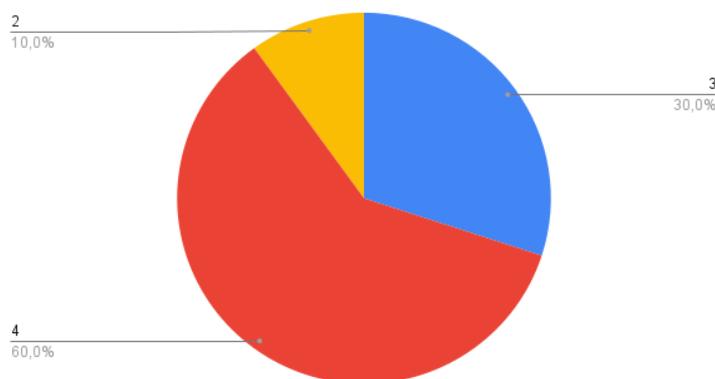
Diagram lingkaran pada Gambar 13 di atas menunjukkan bahwa 45,5% responden setuju dan 54,5% dari 20 responden sangat setuju bahwa sensitifitas suara yang terdeteksi pada *software* akurat.

Tabel 10. Akumulasi Nilai Evaluasi Software - 2

Nilai	Jumlah Pertanyaan Software - 2	Responden Evaluasi	Akumulasi Nilai Pertanyaan Evaluasi Sof- tware - 2
1	0		0
2	1		2
3	6		18
4	13		52
Total	20		72

$$Y = 20 * 4 = 80, \text{ Indeks Evaluasi Software - 1(\%)} = 7280 * 100 = 90\%$$

Tabel tuning yang diberikan sangat membantu



Gambar 14. Diagram Lingkaran Evaluasi Software-2

Diagram lingkaran pada Gambar 14 di atas menunjukkan bahwa 5% responden tidak setuju, 30% responden setuju, dan 65% dari 20 responden sangat setuju bahwa tabel *tuning* yang diberikan di dalam *software* sangat membantu.

Tabel 11. Akumulasi Nilai Evaluasi Software - 3

Nilai	Jumlah Responden	Akumulasi Nilai Pertanyaan Evaluasi Software - 3
1	0	0
2	1	2
3	8	24
4	11	44
Total	20	70

$Y = 20 * 4 = 80$, Indeks Evaluasi Software – 1(%) = $70/80 * 100 = 87,5\%$



Gambar 15. Diagram Lingkaran Evaluasi Software-3

Diagram lingkaran pada Gambar 15 di atas menunjukkan bahwa 5% responden tidak setuju, 40% responden setuju, dan 55% dari 20 responden sangat setuju bahwa hasil yang saya dapatkan saat melakukan *tuning* akurat.

Tabel 12. Akumulasi Nilai Evaluasi Software - 4

Nilai	Jumlah Responden	Akumulasi Nilai Pertanyaan Evaluasi Software - 4
1	0	0
2	2	4
3	5	15
4	13	52
Total	20	71

$Y = 20 * 4 = 80$, Indeks Evaluasi Software – 1(%) = $71/80 * 100 = 88,75\%$



Gambar 16. Diagram Lingkaran Evaluasi Software-4

Diagram lingkaran pada Gambar 16 di atas menunjukkan bahwa 10% responden tidak setuju, 25% responden setuju, dan 65% dari 20 responden sangat setuju bahwa *user* merasa terbantu dengan *Software "Tune Your Guitar"* untuk melakukan *tuning* terhadap gitarnya.

Dengan data-data di atas, menunjukkan bahwa *User Experience* dari aplikasi ini mayoritas sangat baik. Pengguna merasa kemudahan dalam penggunaan *software* ini, dan hasil yang ditampilkan membantu *User* untuk memahami hasil *tuning* yang mereka lakukan. Jika ada perbaikan yang perlu dilakukan adalah dari sisi *theme* dari *web* yang dibuat sehingga lebih *attractive* dan *interactive*.

4. Simpulan

Berdasarkan dari penulisan yang telah dilakukan, penulis dapat menarik kesimpulan dari hasil penulisan yang didapatkan, proses perancangan dan pengembangan perangkat lunak ini dapat membantu dalam mempelajari dan mengetahui nilai frekuensi dan juga hasil notasi yang dihasilkan dari instrumen ataupun vokal suara. Dalam penelitian ini, penulis telah berhasil merancang dan mengembangkan sebuah perangkat lunak *tuning* gitar yang dapat mendeteksi suara secara *real-time* dengan menggunakan *algoritma fourier transformation*. Perangkat lunak "*Tune Your Guitar*" ini berhasil untuk mengukur akurasi kemampuan *algoritma fourier transformasi* dalam menghasilkan nilai frekuensi dan nama notasi yang tepat dengan akurat.

Saran yang bisa dilakukan selanjutnya dari penelitian ini adalah peneliti dapat melakukan perbandingan algoritma selain dengan *Algoritma Fourier Transform (FT)* dengan algoritma lainnya dalam mendeteksi nada, selain itu juga bisa menggunakan alat musik yang lain, selain gitar, apakah memang *Algoritma Fourier Transform (FT)* dapat mendeteksi serta memiliki keakuratan yang sama.

Pustaka

- [1] A. Sumiah and D. T. Lesmana, "Rancang bangun aplikasi sinkronisasi nada pada tuning gitar dengan menggunakan algoritma fast fourier transform berbasis android," *Jurnal Teknologi dan Manajemen Informatika*, vol. 6, no. 1, 2021, <https://journal.uniku.ac.id/index.php/jejaring> [Online].
- [2] M. A. et al., "Identifikasi akor gitar menggunakan algoritma fast fourier transform (fft) berbasis android," in *Prosiding Seminar Informatika Aplikatif Polinema*, 2019.
- [3] E. T. S. R. Basatha and A. H. Dermawan, "Perancangan dan pembuatan aplikasi pembelajaran gitar berbasis android menggunakan metode fast fourier transform," *Teknika*, vol. 10, no. 3, pp. 189–198, Oct. 2021. doi: 10.34148/teknika.v10i3.404.
- [4] N. Hidayat and K. Hati, "Penerapan metode rapid application development (rad) dalam rancang bangun sistem informasi rapor online (siraline)," *Jurnal Sistem Informasi*, vol. 10, no. 1, 2021.

- [5] I. Angelica and C. Nas, "Design ui / ux prototype aplikasi pemesanan produk dimskuy berbasis mobile dengan menggunakan figma," *Jurnal Manajemen Sistem Informasi*, vol. 1, no. 1, 2023.
- [6] I. P. C. T. Pratala, E. M. Asyer and A. Saifudin, "Pengujian white box pada aplikasi cash flow berbasis android menggunakan teknik basis path," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, p. 111, Jun. 2020. doi: 10.32493/informatika.v5i2.4713.
- [7] J. G. F. A. TMS Mulyana, IGN Suryantara, "Natural disaster event mapping in west jawa using k-means algorithm," *Jurnal Algoritma, Logika, dan Komputasi*, vol. 5, no. 2, 2023.
- [8] T. P. D. Wira and R. Andriani, "Unified modelling language (uml) dalam perancangan sistem informasi permohonan pembayaran restitusi sppd," *Jurnal Teknoif Teknik Informatika*, vol. 7, no. 1, 2019.
- [9] K. S. H. et al., "Kajian pengendalian mutu produk tuna loin precooked frozen menggunakan metode skala likert di perusahaan pemebukan tuna," *Authentic Research of Global Fisheries Application Journal*, vol. 2, no. 1, 2020.
- [10] F. Adikara, "Penyuluhan mengenai tantangan revolusi industri 4.0 di bidang pendidikan," *Journal Abdimas*, vol. 6, no. 2, 2020.
- [11] D. S. R. Jiwanda and W. Kurniawan, "Rancang bangun sistem pengkalibrasi nada dan pengkonversi nada menjadi akor pada instrumen gitar dengan labview," 2018, <http://j-ptiik.ub.ac.id> [Online].
- [12] S. F. Adikara and R. Prastya, "Penerapan metode organization goal-oriented requirements engineering (ogore) untuk pembangunan sistem pendaftaran klinik fisioterapi," *Jurnal Edukasi & Pendidikan Informatika*, vol. 6, no. 3, 2020.