

IDENTIFIKASI AKOR GITAR MENGGUNAKAN ALGORITMA *FAST FOURIER TRANSFORM* (FFT) BERBASIS *ANDROID*

Mungki Astiningrum¹, Rosa Andrie Asmara², Novy Dhatika Pramikasuri³

^{1,2,3}Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang

¹mungki.polinema@gmail.com, ²rosa_andrie@polinema.ac.id, ³novydhatika3011@gmail.com

Abstrak – Alat musik gitar adalah salah satu instrumen musik yang banyak dimainkan, namun tidak semua orang paham bagaimana cara memainkan gitar dengan benar. Akor gitar memiliki peran penting untuk memainkan gitar karena dengan akor yang tepat dapat menghasilkan sebuah nada untuk mengiringi sebuah lagu. Pada skripsi ini, penulis merancang sebuah aplikasi identifikasi akor gitar berbasis android yang menggunakan algoritma *Fast Fourier Transform* yang berfungsi untuk menghitung transformasi *Fourier Diskrit* dengan cepat dan efisien, serta melakukan proses pencocokan frekuensi pada frekuensi akor inputan untuk mengetahui frekuensi nada akor gitar tersebut sudah tepat, namun penggunaan aplikasi sangat tidak optimal saat digunakan di keramaian.

Pengujian akurasi aplikasi dilakukan dengan cara menguji aplikasi dengan gitar akustik, hasil tertinggi dari pengujian tersebut sebesar 89,29%. Dari hasil pengujian akurasi tersebut menunjukkan bahwa aplikasi identifikasi akor dapat membantu pengguna untuk belajar memainkan akor gitar.

Kata Kunci : frekuensi, FFT, HPS

I. PENDAHULUAN

Alat musik gitar adalah salah satu instrumen musik yang banyak dimainkan, namun tidak semua orang faham bagaimana cara memainkan gitar dengan benar. Dalam memainkan gitar dibutuhkan banyak pengetahuan secara teori dan teknik. Contoh pengetahuan secara teori dalam memainkan gitar yaitu tentang pengetahuan tangga nada gitar dan bentuk akor gitar. Nada adalah suara atau bunyi yang tersusun dan beraturan. Susunan nada beraturan yang memiliki tingkatan frekuensi tertentu membentuk suatu tangga nada. Tangga nada ini berfungsi sebagai acuan akor gitar dalam memainkan atau membuat lagu.

Akor gitar memiliki peran penting untuk memainkan gitar karena dengan akor yang tepat dapat menghasilkan sebuah nada untuk mengiringi sebuah lagu. Untuk menentukan bentuk akor gitar mayor dan minor dapat dicari menggunakan suatu pola perhitungan pada tangga nada gitar, namun tidak semua orang dapat menentukan akor gitar dengan tepat dibutuhkan pembelajaran yang lebih dengan mengikuti kursus gitar. Membutuhkan waktu dan biaya yang tidak sedikit untuk mengikuti kursus gitar.

Aplikasi identifikasi akor gitar yang akan dibuat tersebut menggunakan algoritma *Fast Fourier Transform* yang berfungsi untuk menghitung transformasi *Fourier Diskrit* dengan cepat dan efisien, serta melakukan proses pencocokan frekuensi pada frekuensi akor inputan untuk mengetahui frekuensi nada akor gitar tersebut sudah tepat. Dari hasil penelitian ini didapatkan hasil berupa aplikasi identifikasi akor gitar yang mampu membantu proses pembelajaran pengguna untuk mempermudah pengenalan akor gitar dengan baik dan informasi tentang tangga nada akor yang teridentifikasi tersebut.

II. TINJAUAN PUSTAKA

A. Nada dan Frekuensi

Nada adalah bunyi yang memiliki frekuensi yang teratur atau jumlah getaran pada setiap detiknya sama. Nada dan frekuensi dapat dilihat pada gambar 1. Nada adalah bunyi yang memiliki frekuensi yang teratur atau jumlah getaran pada setiap detiknya sama.

B. Akor

Chord atau akor secara umum dapat diartikan sebagai suatu rangkaian nada-nada yang tersusun secara teratur dari sebuah tangga nada dan bisa mewakili tangga nada tersebut. Gambar 1 Frekuensi Akor Gitar berisi data notasi dan frekuensi pembentuk akor gitar. Untuk penulisan notasi kotak berwarna kuning dan angka tebal (*bold*) bertujuan untuk mengetahui frekuensi senar pembentuk akor gitar.[1]

TABEL 1. NADA PADA GITAR BESERTA BESAR FREKUENSI

Nama Nada	Frekuensi (Hz)			
C	131	262	523	1047
D	147	294	587	1175
E	82	165	330	659
F	87	175	349	698
G	98	196	392	784
A	110	220	440	880
B	123	247	493	988

TABEL 2. FREKUENSI AKOR GITAR

Akor	Frekuensi Nada Pembentuk Akor Dasar Aplikasi					
	Senar 1	Senar 2	Senar 3	Senar 4	Senar 5	Senar 6
C Mayor	E5 = 659.25 Hz	C5 = 523.25 Hz	G4 = 392.00 Hz	E4 = 329.63 Hz	C4 = 261.63 Hz	E3 = 164.81 Hz
C Minor	G5 = 783.99 Hz	D#5 = 622.25 Hz	C5 = 523.25 Hz	G4 = 392.00 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
D Mayor	F#5 = 739.99 Hz	D5 = 587.33 Hz	A4 = 440.00 Hz	D4 = 293.66 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
D Minor	F5 = 698.46 Hz	D5 = 587.33 Hz	A4 = 440.00 Hz	D4 = 293.66 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
E Mayor	E5 = 659.25 Hz	B4 = 493.88 Hz	G#4 = 415.30 Hz	E4 = 329.63 Hz	B3 = 246.94 Hz	E3 = 164.81 Hz
E Minor	E5 = 659.25 Hz	B4 = 493.88 Hz	G4 = 392.00 Hz	E4 = 329.63 Hz	B3 = 246.94 Hz	E3 = 164.81 Hz
F Mayor	E5 = 659.25 Hz	C5 = 523.25 Hz	A4 = 440.00 Hz	F4 = 349.23 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
F Minor	E5 = 659.25 Hz	C5 = 523.25 Hz	A4 = 440.00 Hz	F4 = 349.23 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
G Mayor	G5 = 783.99 Hz	B4 = 493.88 Hz	G4 = 392.00 Hz	D4 = 293.66 Hz	B3 = 246.94 Hz	G3 = 196.00 Hz
G Minor	G5 = 783.99 Hz	D5 = 587.33 Hz	A#4 = 466.16 Hz	F4 = 349.23 Hz	C4 = 261.63 Hz	G3 = 196.00 Hz
A Mayor	E5 = 659.25 Hz	C#5 = 554.37 Hz	A4 = 440.00 Hz	E4 = 329.63 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
A Minor	E5 = 659.25 Hz	C5 = 523.25 Hz	A4 = 440.00 Hz	E4 = 329.63 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
B Mayor	F#5 = 739.99 Hz	D#5 = 622.25 Hz	B4 = 493.88 Hz	F#4 = 369.99 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz
B Minor	F#5 = 739.99 Hz	D5 = 587.33 Hz	B4 = 493.88 Hz	F#4 = 369.99 Hz	A3 = 220.00 Hz	E3 = 164.81 Hz

C. Pengolahan Sinyal

Sinyal dapat dibagi menjadi dua jenis yaitu sinyal analog (continue) dan sinyal diskrit (digital). Komputer hanya bisa membaca sinyal diskret atau biner sementara di dunia nyata segala sesuatunya secara fisis berupa kuantitas analog. Kuantitas analog ini diubah menjadi besaran listrik (nilai tegangan atau arus yang setara) menggunakan transducer sebelum masuk rangkaian ADC untuk diubah menjadi sinyal digital.[2]

D. Android

Tujuan pembuatan sistem operasi ini adalah untuk menyediakan platform yang terbuka, yang mempermudah akses internet menggunakan telepon seluler. Android juga dirancang untuk memudahkan pengembang membuat aplikasi dengan batasan yang minim sehingga kreativitas pengembang menjadi lebih berkembang.[3]

E. Windowing

Dalam domain frekuensi, filter sifatnya adalah terbatas yaitu hanya melakukan sinyal pada frekuensi tertentu. Oleh karena itu untuk membatasi panjang filter dalam domain waktu digunakan windowing.[2]

F. FFT

Transformasi Fourier ditujukan untuk mengubah domain waktu dari sebuah sinyal ke domain frekuensi. Pada algoritma FFT deretan angka yang berada dalam domain waktu $f(x)$ diubah kedalam domain frekuensi $f(u)$ dengan persamaan.[4]

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

N adalah point transformasi fourier yang digunakan.

G. HPS

HPS (*Harmonic Product Spectrum*) merupakan suatu metode yang berfungsi untuk melihat frekuensi dasar yang terdapat pada sinyal input. Secara matematis HPS dapat dirumuskan seperti berikut :

$$(k) = \left(\prod (n k) \right) N n=1 \quad (2)$$

HPS adalah hasil spektrum harmonik, k adalah indeks frekuensi spektrum harmonik, Y adalah besar spektrum pada frekuensi positif, dan N merupakan jumlah harmonik yang digunakan.[5]

III. IMPLEMENTASI DAN PENGUJIAN

A. Implementasi Aplikasi

Implementasi tampilan aplikasi identifikasi akor yang berisi akor mayor dan minor gitar. Tampilan secara rinci terdiri dari tampilan splash screen, tampilan home, tampilan daftar akor, tampilan notifikasi berhasil, gagal dan tampilan mainkan akor gitar. Berikut adalah tampilan aplikasi identifikasi akor gitar.



Gambar 1. Tampilan splash screen



Gambar 2. Tampilan home

Gambar 1. *splash screen* adalah tampilan yang pertama kali muncul saat aplikasi dibuka. Tampilan *splash screen* akan muncul dalam beberapa detik lalu akan secara otomatis berubah menjadi tampilan *home*.

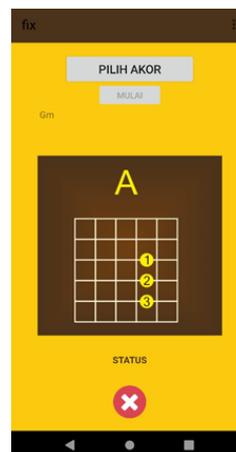
Gambar 2. *home* berisi button untuk menampilkan daftar gambar akor menuju tampilan identifikasi akor gitar. Gambar 3. identifikasi akor akan berjalan ketika button start ditekan. Aplikasi merekam selama sepuluh detik untuk mengidentifikasi akor. Tampilan identifikasi akor menghasilkan dua notifikasi yaitu notifikasi berhasil dengan simbol centang dan notifikasi tidak berhasil bersimbol silang. Saat simbol centang muncul akan terdengar suara notifikasi berhasil sedangkan saat simbol silang tidak memiliki suara notifikasi.



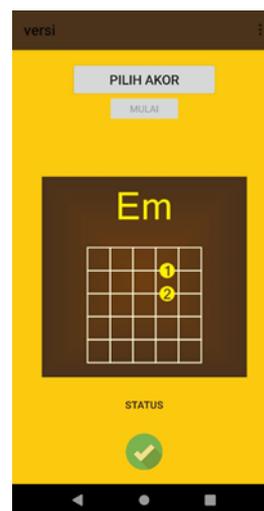
Gambar 3. Tampilan daftar pilihan akor



Gambar 4. Tampilan perintah mainkan gitar



Gambar 5. Tampilan Notifikasi Gagal



Gambar 6. Tampilan Notifikasi Berhasil

Gambar 4 adalah Tampilan notifikasi text berisi “Mainkan gitar sesuai gambar akor!” akan muncul ketika aplikasi tidak mengidentifikasi adanya suara yang tertangkap. Selain tampilan notifikasi text “Mainkan / petik gitar sesuai gambar akor!” ada juga dua notifikasi yaitu notifikasi berhasil dengan simbol centang yaitu Gambar 6 dan notifikasi tidak berhasil bersimbol silang pada Gambar 5. Saat simbol centang muncul akan terdengar suara notifikasi berhasil sedangkan saat simbol silang akan muncul teks yang berisi kesimpulan akor yang ditangkap oleh aplikasi.

B. Perhitungan FFT manual

Contoh data yang digunakan untuk perhitungan FFT manual adalah waktu $f(x)$ bernilai 0, 1, 2, 3 dan 4 sedangkan data (u) berisi 2, 3, 4 dan 4. Perhitungan FFT secara manual dengan menggunakan rumus :

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

Berdasarkan rumus tersebut maka diketahui nilai N pada contoh data adalah 4 data. Hasil perhitungan FFT secara manual dijelaskan Tabel 3 dan data perhitungan pada Tabel 2.

TABEL 2. DATA PERHITUNGAN FFT MANUAL

Waktu f(x)	Data (u)
0	2
1	3
2	4
3	4

TABEL 3. HASIL PERHITUNGAN FFT SECARA MANUAL

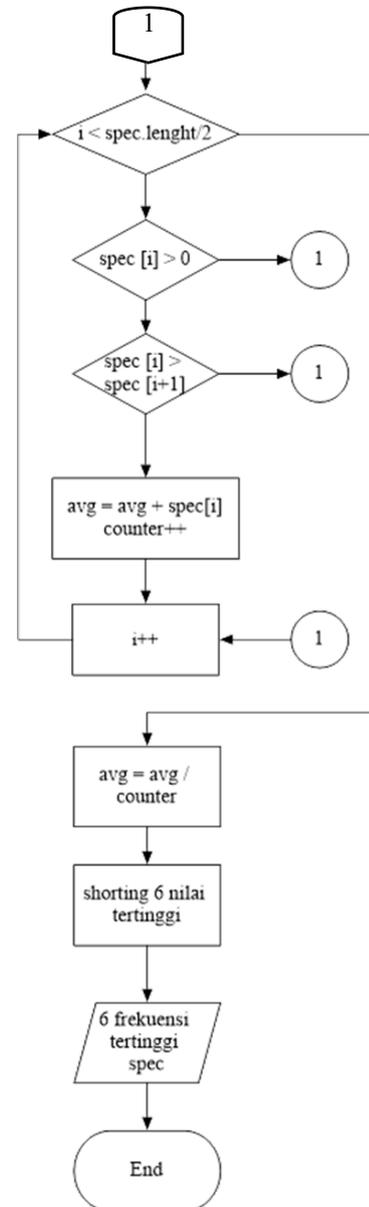
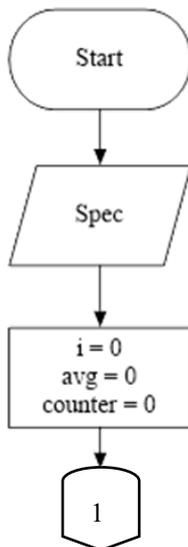
No	f(x)	Perhitungan	f(u)
1	f(0)	$2(1-j(0))+3(1-j(0))+4(1-j(0))+4(1-j(0))$	13
2	f(1)	$2+3(0-j(1))+4(-1-j(0))+4(-0-j(-1))$	$(-2)+j$
3	f(2)	$2+3(-1-j(0))+4(1-j(-0))+4(-1-j(0))$	-1
4	f(3)	$2(\cos(2*\pi*3*0/4)-j \sin(2*\pi*3*0/4))$	
		$3(\cos(2*\pi*3*1/4)-j \sin(2*\pi*3*1/4))$	
		$4(\cos(2*\pi*3*2/4)-j \sin(2*\pi*3*2/4))$	
		$4(\cos(2*\pi*3*3/4)-j \sin(2*\pi*3*3/4))$	
		$2(1-j(0))+3(-0-j(-1))+4(-1-j(0))+4(0-j)$	$(-2)-j$

C. Hasil perhitungan FFT manual

Pada Tabel 3. Hasil perhitungan FFT secara manual di dalam proses perhitungan secara rinci terdapat pada perhitungan ke empat, baris yang berwarna abu-abu tersebut adalah proses penjumlahan dari hasil perhitungan ke empat baris sebelumnya. Pada kolom f(u) merupakan hasil dari perhitungan FFT secara manual.

D. Implementasi Harmonic Product Spetrum (HPS)

Perhitungan HPS dilakukan setelah proses FFT selesai. HPS digunakan untuk menemukan titik puncak dari suatu gelombang. Perhitungan HPS pada aplikasi ini mencari enam titik puncak karena terdapat enam senar pada gitar yang dipetik. Keenam senar gitar tersebut memiliki nilai frekuensi tersendiri yang akan ditemukan menggunakan HPS. Gambar 7. Flowchart HPS adalah gambar yang menjelaskan flowchart HPS yang berjalan pada aplikasi.



Gambar 7. Flowchart HPS

E. Hasil pengujian

TABEL 4. HASIL PENGUJIAN APLIKASI DENGAN GITAR AKUSTIK

No	Akor	Hasil benar	Hasil salah	Akurasi %	Error %
1	C mayor	8	2	80 %	20 %
2	C minor	10	0	100%	0 %
3	D mayor	10	0	100%	0 %
4	D minor	10	0	100%	0 %
5	E mayor	8	2	80 %	20 %
6	E minor	8	2	80 %	20 %
7	F mayor	10	0	100%	0 %
8	F minor	10	0	100%	0 %
9	G mayor	6	4	60%	40%
10	G minor	10	0	100%	0 %
11	A mayor	8	2	80 %	20 %
12	A minor	9	1	90%	10%
13	B mayor	9	1	90%	10%
14	B minor	9	1	90%	10%
	TOTAL	125	15	89,29%	10,71%

TABEL 5. HASIL PENGUJIAN APLIKASI DENGAN VIRTUAL GITAR

No	Akor	Hasil benar	Hasil salah	Akurasi %	Error %
1	C mayor	8	2	80 %	20 %
2	C minor	9	1	90%	10%
3	D mayor	10	0	100%	0 %
4	D minor	10	0	100%	0 %
5	E mayor	9	1	90%	10%
6	E minor	9	1	90%	10%
7	F mayor	10	0	100%	0 %
8	F minor	10	0	100%	0 %
9	G mayor	10	0	100%	0 %
10	G minor	7	3	70%	30%
11	A mayor	9	1	90%	10%
12	A minor	10	0	100%	0 %
13	B mayor	9	1	90%	10%
14	B minor	10	0	100%	0 %
	TOTAL	130	10	92,86%	7,14%

TABEL 6. HASIL PENGUJIAN APLIKASI DENGAN CHORD DETECTOR

No	Akor	Hasil benar aplikasi	Hasil benar chorddetector	Akurasi aplikasi %	Akurasi chorddetector %
1	C mayor	8	5	80%	50%
2	C minor	10	9	100%	90%
3	D mayor	10	9	100%	90%
4	D minor	10	2	100%	20%
5	E mayor	6	8	60%	80%
6	E minor	9	6	90%	60%
7	F mayor	10	9	100%	90%
8	F minor	10	3	100%	30%
9	G mayor	6	10	60%	100%
10	G minor	10	9	100%	90%
11	A mayor	8	10	80%	100%
12	A minor	10	7	100%	70%
13	B mayor	8	7	80%	70%
14	B minor	10	7	100%	70%
	TOTAL	135	101	96,43%	72,14%

Pada Tabel 4 Hasil percobaan aplikasi dengan gitar akustik menghasilkan presentase akurasi serta error keempat belas akor dengan hasil keseluruhan rata – rata akurasi 89,29% dan error 10,71%. Hasil pengujian aplikasi dengan virtual gitar tertera pada Tabel 5 Hasil pengujian aplikasi dengan virtual gitar. Nilai presentase akurasi dari pengujian tersebut adalah 92,86% dengan hasil nilai error 7,14%. Hasil pengujian perbandingan aplikasi dan aplikasi chord detector dengan virtual gitar tertera pada Tabel 6 Hasil percobaan aplikasi dengan chord detector dengan nilai presentase akurasi aplikasi dari pengujian tersebut adalah 96,43% dan nilai akurasi aplikasi chord detector 72,14%.

IV. KESIMPULAN

- Aplikasi identifikasi akor dapat membantu pengguna untuk belajar memainkan akor gitar karena dapat dilihat dari tingkat akurasi aplikasi dengan gitar akustik sebesar 89,29%.

- Aplikasi dapat mendeteksi nada akor gitar dan menampilkan gambar petunjuk bentuk akor gitar.
- Pengujian dengan aplikasi chord detector menunjukkan perbedaan hasil yaitu 96,43% untuk aplikasi dan 72,14% aplikasi chord detector. Perbedaan hasil banyak ditemukan pada pengujian akor minor. Sebagai contoh saat gitar virtual memainkan akor E minor pada pengujian menggunakan aplikasi chord detector hasil yang terdeteksi adalah akor E mayor.

REFERENSI

- [1] “The Notes of the Guitar,” p. 2007, 2007.
- [2] N. Pradipta, “Implementasi Algoritma FFT (Fast Fourier Transform) Pada Digital Signal Processor (DSP) TMS320C542,” *Implementasi Algoritma. FFT (Fast Fourier Transform. Pada Digit. Signal Process. TMS320C542*, no. Universitas Diponegoro, pp. 1–7, 2011.
- [3] A. Danielsson and O. Bälter, “KTH-Royal Institute of Technology Comparing Android Runtime with native: Fast Fourier Transform on Android,” 2017.
- [4] G. Sewell, “The Fast Fourier Transform,” *Comput. Methods Linear Algebr.*, pp. 210–224, 2014.
- [5] K. Muludi, A. Frank, and S. F. B. Loupatty, “Chord Identification Using Pitch Class Profile Method with Fast Fourier Transform Feature Extraction,” vol. 11, no. 3, pp. 139–144, 2014.