

PERANCANGAN DAN PEMBUATAN APLIKASI PENGENAL NADA DASAR PADA FILE LAGU MONOPHONIC

Yanuar Christian Ardianto¹, Liliana, M.Eng.², Gregorius Satia Budhi, M.T.³

Program Studi Teknik Informatika, Fakultas Teknologi Industri Universitas Kristen Petra
Jalan Siwalankerto 121- 129 . Telp. 031 - 8439040

Email: Sukrisno0101@gmail.com, Liliana@petra.ac.id, greg@petra.ac.id

ABSTRAK

Musik telah digemari di seluruh dunia. Untuk belajar musik, banyak orang yang memerlukan kursus untuk mempelajarinya. Sedangkan untuk orang yang mempelajari sendiri sering terjadi kesulitan saat menentukan melodi dan nada dasar.

Aplikasi ini menyediakan fitur untuk mengenali nada dasar dan melodi dari sebuah file lagu *monophonic* dan menampilkannya dalam bentuk tabel. Adapun aplikasi ini dirancang untuk beroperasi pada *operating system Windows*.

Berdasarkan hasil pengujian, aplikasi ini dapat berjalan dengan baik pada file rekaman alat musik yang menghasilkan nada yang konstan dan kualitas suara yang baik.

Kata Kunci

Monofonik, Nada Dasar, Melodi

ABSTRACT

Music has been popular all over the world. To learn music, many people need take a course to study it. As for people who study himself facing a lot of difficulties when determining the basic tone and melodies.

This application provides features to recognize the basic tone and melodies from a monophonic music file and display it in a table. This application is designed to work on Window operating system.

Based on testing results, this application can run properly on file recording instruments that produce a constant tone and good sound quality.

Keywords

Monophonic, Basic Tone, Melodies

1. PENDAHULUAN

Musik telah berkembang dan digemari oleh seluruh orang di dunia. Berkembang dari sebuah pertunjukan langsung, menuju rekaman yang disimpan dalam piringan hitam, *compact disc*, dan *audio file*. [1] Selain mendengarkan musik, orang-orang tertarik untuk belajar memainkan alat musik. Dengan mengambil kursus atau belajar dari orang lain, orang-orang dapat mempelajari alat-alat musik. Sebagian orang tidak memiliki cukup waktu atau uang untuk mengambil kursus musik, sehingga mereka mempelajari alat musik sendiri. Bagi pemula eringkali dalam mempelajari rekaman lagu terdapat kesulitan dalam menentukan nada dasar dan melodi dari sebuah lagu. Berdasarkan permasalahan di atas, dibuatlah sebuah perancangan dan pembuatan aplikasi untuk membantuk orang mengetahui nada dasar dan melodi dari sebuah *audio file*. Dalam pengenalan nada dasar, diperlukan pengenalan melodi dari

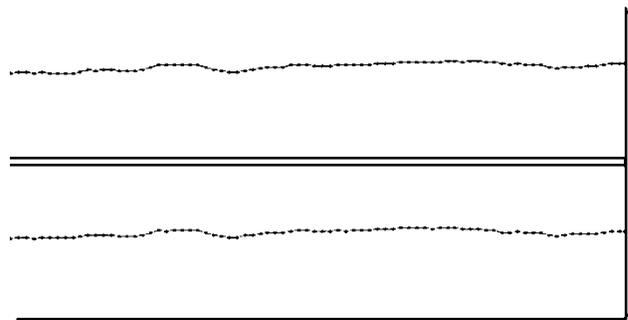
sebuah file lagu. Algoritma *pitch detection* yang digunakan adalah *Harmonic Product Spectrum*. Dengan aplikasi ini diharapkan orang yang awam dalam bermain musik bisa lebih mudah mempelajari musik dan diharapkan aplikasi ini dapat dikembangkan lebih lagi.

2. PITCH DETECTION ALGORITHM

Algoritma *Pitch detection* yang akan digunakan adalah *Harmonic Product Spectrum*. *Harmonic Product Spectrum* (HPS) memerlukan beberapa proses pengolahan dari *audio file* yaitu *windowing* dan *Fast Fourier Transform*.

2.1 Digital Audio

Digital audio adalah rekaman digital dari gelombang suara analog. Sebuah *file* yang menyimpan suara terdiri dari ribuan angka yang mencatat amplitudo dari gelombang suara.



Gambar 1. Bentuk Waveform dari Sebuah Audio File

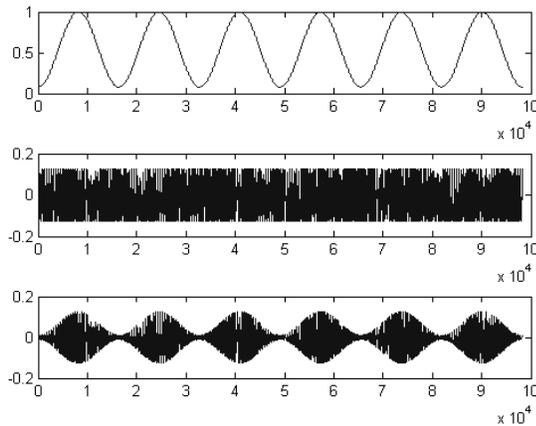
Pada gambar 2 terlihat titik-titik kecil yang berisikan data informasi gelombang yang disebut sebagai *sample*. Ada beribu-ribu *sample* pada setiap detik dari sebuah lagu. Jumlah *sample* per detik pada sebuah file lagu disebut dengan *sampling rate* dan biasanya memiliki standar sekitar 44.100 *sample* per detik atau ditulis dengan 44.100 Hz. Semakin tinggi jumlah *sample* per detik maka semakin banyak data poin setiap detiknya, yang artinya file audio dengan kualitas lebih tinggi. *Sampling rate* yang lain yang umum adalah 48 KHz, 96 KHz, 192 KHz, dan 8 KHz. Pada sebuah lagu dengan 44.100 Hz *sampling rate*, kita akan memiliki 44.100 data point setiap detiknya. Tetapi bukan setiap data point yang menyebabkan kita bisa mendengar suara, tetapi pergantian dari ribuan data point yang berulang-ulang menyebabkan kita bisa mendengar suara. [2]

Jumlah getaran dari gelombang per detik disebut dengan *fundamental frequency*. Jika sebuah gelombang sinus bergetar sebanyak 440 kali dalam satu detik, maka akan kita dengar dengan nada 'Concert A(440 Hz)'. [2]

2.2 Hamming Window

Ketika kita menggunakan FFT untuk mengukur frekuensi dari data, kita harus menentukan dasar analisa pada sebuah data

yang pasti. FFT mengasumsikan bahwa data yang tetap adalah sebuah periode dari *period signal*. Bagi FFT *time-domain* dan *frequency-domain* adalah perputaran yang sama, sehingga titik terakhir dari *time waveform* dianggap berhubungan. Sehingga keterbatasan dalam *sampling record* menghasilkan *waveform* yang terpotong dengan perbedaan karakter dari sinyal aslinya, dan keterbatasan ini membuat perubahan yang tajam dalam pengukuran data. Untuk mengurangi efek perubahan ini, kita bisa menggunakan fungsi Window untuk sinyal dalam *time-domain* sebelum diproses menuju FFT. aman selalu digital dengan A-D *converter* untuk mengubah sinyal analog menjadi digital sebelum disimpan dalam komputer. Proses komputasi spektrum dari sinyal yang telah direkam adalah menggunakan *Fast Fourier Transform*. [3].



Gambar 2 Hasil dari Hamming Window

2.3 Fast Fourier Transform

Dalam spektrum *time-domain*, kita terbatas hanya pada sinyal sinusoids dan sinyal *periodic*. Untuk memperluas cara kita untuk memproses sinyal, kita perlu mengubah dari *time domain* ke dalam bentuk *frequency spectrum* yang akan diaplikasikan kepada semua sinyal (t). Dengan cara ini kita akan bisa: (1) menjelaskan sebuah nosi yang tepat dari *bandwidth* untuk sebuah sinyal, (2) menjelaskan cara kerja dari sistem komunikasi yang modern dimana mungkin untuk mengirimkan banyak sinyal bersamaan dengan membagi *bandwidth* yang ada dan, (3) menjelaskan operasi *filtering* yang dibutuhkan untuk memisahkan sinyal dalam sebuah *frequency-shared systems*. Ada banyak aplikasi lainnya dari *Fourier Transform* yang bermanfaat, jadi bisa dikatakan bahwa analisa Fourier memberikan sebuah bahasa yang dibutuhkan untuk mendefinisikan dan mendesain sistem teknik modern.[3]

Analisa spektrum dengan menggunakan algoritma *Fast Fourier Transform* (FFT) adalah yang paling sering digunakan oleh teknisi atau ilmuwan dalam pengaplikasian *Fourier Transform*.

$$\sum_{n=0}^{N-1} a_n e^{-2\pi i n k / N} = \sum_{n=0}^{N/2-1} a_{2n} e^{-2\pi i (2n) k / N} \quad (1)$$

$$+ \sum_{n=0}^{N/2-1} a_{2n+1} e^{-2\pi i (2n+1) k / N} \quad (2)$$

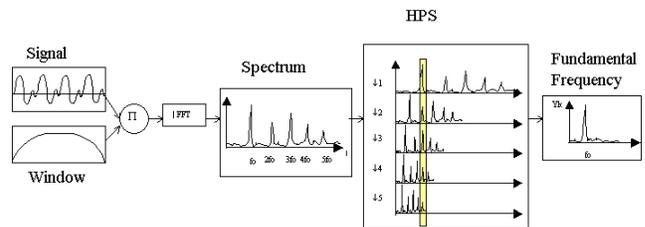
$$= \sum_{n=0}^{N/2-1} a_n^{even} e^{-2\pi i n k / (\frac{N}{2})} \quad (3)$$

$$+ e^{-2\pi i k / N} \sum_{n=0}^{N/2-1} a_n^{odd} e^{-2\pi i n k / (\frac{N}{2})} \quad (4)$$

Dalam menganalisa hasil dari FFT, kita memerlukan analisa *time-frequency*. Di dalamnya kita akan menganalisa sebuah sinyal yang sangat panjang dengan melakukan banyak FFT pendek, yang kemudian digabungkan menjadi sebuah gambar *grayscale*, yang dinamakan *spectrogram*.

2.4 Harmonic Product Spectrum

Dasar dari pengenalan nada dasar adalah *pitch detection algorithm*. *Harmonic Product Spectrum* merupakan metode *pitch detection algorithm* yang paling simpel dan bekerja dengan baik dalam berbagai kondisi[4].



Gambar 3.Flowchart HPS

Algoritma HPS mengukur *maximum coincidence* dari *harmonic* berdasarkan persamaan (1) untuk setiap *spectral frame* $X(w)$.

$$Y(w) = \prod_{r=1}^R |X(wr)| \quad (1)$$

$$\hat{Y} = \max_{w_i} \{Y(w_i)\} \quad (2)$$

R adalah jumlah dari *harmonic* yang akan digunakan (contoh $R=5$), dan frekuensi w_i ada di antara *fundamental frequencies* yang mungkin. Kemudian dicari nilai maksimum \hat{Y} dari hasil *periodic correlation array* seperti pada persamaan (2).

Octave error sering terjadi pada penentuan hasil dari HPS. *Octave errors* adalah hasil dari HPS lebih tinggi satu oktaf dari *pitch* sebenarnya. Untuk mengatasi kesalahan seperti ini, ditambahkan sebuah *rule*: Jika amplitudo *pitch* yang tertinggi kedua kira-kira setengah dari *pitch* yang terpilih, dan ratio dari amplitudo diatas *threshold* (contoh 0.2 untuk 5 *harmonics*), maka pilih satu oktaf dibawah oktaf yang terpilih.

3. PENGENAL NADA DASAR DAN MELODI

Setelah menemukan *pitch* dari potongan – potongan *audio file*, untuk menentukan nada dasar, perlu merubah informasi *pitch* yang berupa angka menjadi notasi huruf beserta dengan oktaf.

3.1 Pengenal Melodi

Untuk menghitung notasi huruf dari *pitch* yang dihasilkan dari *pitch detection algorithm*, digunakan standar dari frekuensi nada yang biasa digunakan dalam musik yaitu 440 Hz (nada A pada oktaf keempat). Frekuensi dasar dari setiap nada dihitung dengan menggunakan persamaan $f_n = 2^{n/12} * 440$ dimana n adalah jumlah jarak *half step* dari *standard tuning* yaitu 440 Hz. Jika *pitch* hasil sama dengan frekuensi pada nada yang sedang dibandingkan, maka notasi dari *pitch* hasil adalah nada tersebut. Jika *pitch* tidak benar-benar identik dengan frekuensi nada dan *pitch* hasil lebih besar dari frekuensi tersebut, maka dicek dengan nada *half step* berikutnya. Jika berada diantara dua nada tersebut, dilakukan perbandingan jarak antara nada pertama dengan nada kedua. Jarak yang terdekat menentukan nada yang dipilih sebagai hasil.[5]

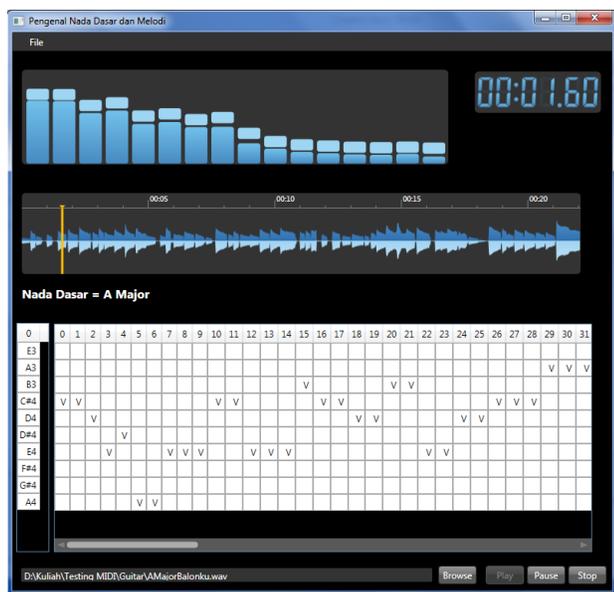
3.2 Menentukan Nada Dasar

Setelah menemukan melodi apa saja dan not apa saja yang terdapat dalam file lagu tersebut, aplikasi ini dapat menentukan nada dasar. Sebuah lagu dalam skala mayor memiliki tiga elemen penting untuk nada dasar. Tiga elemen tersebut adalah do, mi, dan sol. Sedangkan pada skala minor, tiga elemen penting tersebut adalah do, ri (re ditambah setengah), dan sol. [6]

Semua melodi yang disimpan pada hasil array not akan dicek pada setiap dua belas nada dasar dan sebanyak jumlah nada dalam array not. Untuk skala mayor, untuk setiap nada do dari dua belas nada dasar, maka dihitung kemungkinannya sebagai nada dasar. Begitu juga nada re, mi, fa, sol, la, dan si. Tetapi yang memiliki bobot lebih adalah pada do, mi, dan sol sebagai tiga elemen penting dalam membentuk skala mayor. Selain skala mayor, skala minor juga akan dicek pada setiap dua belas nada dasar.

4. PENGUJIAN DAN HASIL

Pengujian dilakukan pada lima alat musik dengan dua belas nada dasar yang berbeda, juga dibedakan antara lagu mayor dan lagu minor. Pengujian dilakukan dengan membandingkan hasil dari melodi dan nada dasar yang dibaca dibandingkan dengan hasil dari aplikasi.



Gambar 4 Tampilan dari Aplikasi Saat Membaca File Lagu

Pengujian dengan alat musik piano elektrik yang memainkan skala mayor sederhana menunjukkan hasil yang sangat baik pada pengenal melodi maupun pada pengenal nada dasar. Begitu juga dengan pengujian pada alat musik gitar akustik pada skala mayor sederhana. Sedangkan pada alat musik seperti biola, bass, serta vokal memiliki akurasi yang buruk akibat kualitas suara saat perekaman. Aplikasi ini sangat dipengaruhi oleh kualitas suara rekaman yang akan diinputkan.

Tabel 1. Tabel Pengujian Alat Musik

Alat Musik	Lagu yang diuji	Akurasi	Akurasi
		Nada Dasar	Melodi
Piano	Skala Mayor	100%	99,50%
	Skala Minor	91,67%	100%
	Balonku(Mayor)	75%	97,32%
	Kadosh(Minor)	50%	99,50%
Gitar	Skala Mayor	83,33%	99,17%
	Skala Minor	83,33%	98,66%
	Balonku(Mayor)	83,33%	99,17%
	Kadosh(Minor)	41,67%	94,84%
Bass	Skala Mayor	66,67%	79,62%
	Skala Minor	66,67%	78,21%
	Balonku(Mayor)	33,35%	79,20%
	Kadosh(Minor)	58,33%	82,70%
Vokal	Skala Mayor	58,33%	81,71%
	Skala Minor	25%	79,62%
	Balonku(Mayor)	0%	72,80%
	Kadosh(Minor)	16,67%	67,49%
Biola	Skala Mayor	57,14%	82,44%
	Balonku(Mayor)	71,43%	86,92%

Dari hasil pengujian dapat dilihat semakin buruk akurasi dari pengenal melodi, maka semakin buruk hasil nada dasar yang dihasilkan. Seperti pada bass, terdapat banyak nada yang seharusnya tidak dimainkan, terbaca oleh aplikasi dan diperhitungkan untuk menentukan nada dasar, sehingga hasil dari nada dasar salah.

Pada lagu kadosh, tidak terdapat melodi yang menjadi elemen dari sebuah skala minor. Sehingga akurasi pengenal nada dasar buruk meskipun memiliki akurasi pengenal melodi yang cukup tinggi seperti pada gitar dan piano.

Pada pengujian terhadap panjang pendeknya sebuah lagu tidak berpengaruh pada hasil nada dasar dan pengenalan melodi. Berikut adalah tabel pengujian terhadap panjang pendeknya sebuah lagu.

Tabel 2. Tabel Pengujian terhadap Durasi Lagu

Lagu yang diuji	Durasi	Nada Dasar Sebenarnya	Nada Dasar Hasil	Akurasi Melodi
Pendek	(00.09)	D Mayor	D Mayor	97,95%
	(00.13)	D Mayor	D Mayor	97,33%
Sedang	(00.58)	D Mayor	D Mayor	97,13%
	(00.49)	D Mayor	D Mayor	98,86%

Pengujian yang ketiga dilakukan terhadap seberapa acak nada yang dimainkan terhadap hasil nada dasar dan melodi.

Tabel 3. Tabel Pengujian terhadap Nada yang acak

Lagu yang diuji		Nada Dasar Sebenarnya	Nada Dasar Hasil	Akurasi Melodi
Simpel	Test 1	D Mayor	D Mayor	97,29%
	Test 2	D Mayor	D Mayor	95,78%
Agak Acak	Test 1	D Mayor	B Mayor	98,37%
	Test 2	D Mayor	E Minor	95,13%
Acak	Test 1	Chromatic	B Minor	95,65%
	Test 2	Chromatic	B Mayor	92,15%

Acaknya nada sangat mempengaruhi hasil dari pengenalan nada dasar, sedangkan pada akurasi melodi bergantung pada kualitas suara dari alat musik yang direkam.

5. REFERENCES

- [1] Benward, Bruce., White, Gary . (1997). *Music in Theory and Practice Volume 1* (Sixth Edition). BenchMark Publishers.
- [2] *Microsoft Corporation* (2013). *Intro to Audio Programming*, Diambil 16 Januari 2013. DOI=<http://blogs.msdn.com/b/dawate/archive/2009/06/22/intro-to-audio-programming-part-1-how-audio-data-is-represented.aspx>
- [3] McClellan, Schafer, Yoder. (2003). *Signal Processing First*. Prentice Hall.
- [4] Middleton, Gareth (2003). *Pitch Detection Algorithm*. Diambil 2 Januari 2013. DOI=<http://cnx.org/content/m11714/latest/>
- [5] Perry, Gabriel, (2006) .*The 12 Notes of Music*, Diambil 18 Januari 2013 DOI=http://www.flupe.com/lessons/freeLessons/01_foundationLessons/L01_12Notes/12_notes.htm
- [6] Wolfe, Joe (2005) *Note Names, MIDI Numbers and Frequencies*. Diambil tanggal 20 Feb 13 DOI= <http://www.phys.unsw.edu.au/jw/notes.html>