

APLIKASI PENGKONVERSI NADA-NADA INSTRUMEN TUNGGAL MENJADI *CHORD* MENGGUNAKAN METODE *PITCH CLASS PROFILE*

Imam Gaffar^{*)}, Achmad Hidayatno, and Ajub Ajulian Zahra

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: imamgaffar@yahoo.co.id

Abstrak

Chord recognition atau pengenalan chord adalah sebuah transkripsi dari suara menjadi chord atau proses pengkonversian sebuah masukan yang berupa file audio menjadi sebuah tanda yang dapat dikenali sebagai suatu huruf atau yang disebut dengan chord. Chord merupakan tiga buah nada atau lebih yang dibunyikan secara bersamaan sehingga menciptakan suatu suara yang harmonis. Chord yang akan dikenali dalam aplikasi ini adalah chord standar yaitu chord mayor. Tujuan dari penelitian ini adalah membuat perangkat lunak yang dapat mengkonversi nada-nada dari instrument tunggal menjadi chord secara otomatis dengan input file audio, yang mana di dalamnya terjadi pemrosesan sinyal audio. Tahapan proses yang dilakukan adalah perekaman audio dari instrument tunggal berupa gitar akustik dan gitar bass, pengambilan data audio, front - end detection, FFT, deteksi puncak frekuensi, dan pengelompokan frekuensi-frekuensi ke dalam nada atau dengan sebutan Pitch Class Profile . Perancangan aplikasi ini dilakukan dengan metode PCP (Pitch Class Profile) yang mana nada akan dideteksi pada frekuensi tertentu berdasarkan pengkelasan nada dan kemudian akan ditampilkan hasil akhir dari sistem tersebut berupa huruf chord beserta chord dialognya. Pembuatan aplikasi ini menggunakan bahasa pemrograman pascal dengan software Delphi 7 dengan didukung komponen-komponennya. Berdasarkan hasil pengujian dan analisis, tingkat keberhasilan dari masing-masing nada untuk menjadi chord pada gitar bass adalah 99,1 % dengan persentase error sebesar 0,84%. Sedangkan untuk gitar akustik, tingkat keberhasilannya sebesar 72,27 % dan persentase error sebesar 27,72%.

Kata Kunci: Chord, Pitch Class Profile, FFT, Instrumet tunggal.

Abstract

Chord recognition is a transcription of voice to be chord or the process of converting an input audio file into a signal that can be recognized as a letter or called a chord. Chord is three or more tones are sounded simultaneously, creating a harmonious sound. Chord will be recognized in this application is standard chord which it is a major chord. The purpose of this thesis is to create software that can convert the pitch from single instrument to be chord automatically with input audio file, which occurs inside the audio signal processing. Stage of the process is the audio recording of a single instrument that is an acoustic guitar and bass guitar, audio data retrieval, front-end detection, FFT, peak detection of FFT frequency and classification of frequencies into a tone or it can be called the Pitch Class Profile. The Implementation of this application is using PCP (Pitch Class Profile) methode which tone that detected on a certain frequency based classes are the tone then it will display the final results of the system in the form of letters chord and chord dialogue. This applications using the programming language Pascal with Delphi 7 supported software components. Based on the test results and analysis, the success rate of each tone to a chord on bass guitar is 99.1% with a percentage error of 0.84%. As for the acoustic guitar, the success rate of 72.27% and the percentage error of 27.72%.

Keyword : Chord, Pitch Class Profile, FFT, Single Intrument.

1. Pendahuluan

Dalam kehidupan sehari-hari, manusia tidak bisa lepas dari yang namanya musik. Di dalam dunia musik terdapat istilah chord yaitu akord atau biasa juga disebut dengan "Kunci". Chord adalah gabungan beberapa nada yang dibunyikan sehingga menciptakan suara harmonis. Dalam

sebuah lagu chord dapat dimainkan atau dibunyikan secara terputus-putus maupun secara bersamaan. Secara garis besar chord dikategorikan menjadi beberapa tipe, antara lain : chord-chord mayor, minor, diminished, augmented, suspended, seventh, dan lain-lain.[9]

Untuk memainkan sebuah lagu pemain musik harus

mengerti dan mencari chord-chord apa saja yang ada pada lagu tersebut. Para pemain musik yang sudah mengerti tentang musik sangat mudah dalam mencari chord-chord yang ada pada sebuah lagu dengan cara mendengarkan lagu tersebut. Dalam hal ini muncul permasalahan bagaimana dengan pemula atau seseorang yang baru mempelajari musik, seorang pemula akan begitu sulitnya dalam mencari chord-chord dalam suatu lagu. Hal ini disebabkan karena belum mengerti tentang nada-nada ataupun kurang pahamnya suatu karakteristik dari suatu chord tersebut.

Fungsi untuk mengenali chord secara otomatis sangat penting dalam beberapa aplikasi, diantaranya seperti sistem musik interaktif, aplikasi informasi musik berdasarkan konten (pencarian contoh-contoh atau tema-tema lagu dalam database audio) serta aplikasi edukasi. Chord recognition atau pengenalan chord adalah sebuah transkripsi dari suara menjadi chord, yang mana dapat diklasifikasikan sesuai dengan karakteristiknya yang berbeda-beda, dari sebuah perbedaan yang sederhana antara chord major dan minor dengan sekomples tipe chord (maj, min, dim, aug, 7th, dan lain-lain). [9]

Pada penelitian ini akan dibangun sebuah aplikasi pengenalan chord secara otomatis, untuk membantu seorang pemula dalam belajar musik atau untuk mengetahui karakteristik chord-chord dalam sebuah lagu. Pada aplikasi ini juga dapat memberikan informasi tentang chord-chord yang dimainkan atau ditekan oleh orang yang ingin belajar musik khususnya untuk alat musik bass dan gitar akustik.

Penelitian ini bertujuan untuk membangun suatu perangkat lunak yang dapat mendeteksi chord secara otomatis yang didalamnya terjadi pemrosesan sinyal audio sehingga audio input akan dikenali oleh sistem, dan menghasilkan output berupa informasi chord dari file audio tersebut berupa huruf beserta chord dialog yang nantinya akan membantu pemula dalam memahami suatu chord.

Adapun batasan masalah dalam penelitian ini diantaranya adalah sebagai berikut : File input yang digunakan adalah lagu bertipe (*.wav), file yang direkam berupa instrumen tunggal, hasil keluaran berupa informasi *chord* berupa huruf beserta *chord* dialognya, *chord* yang dikenali adalah *chord* standar yaitu berupa *chord* Major, Aplikasi tidak *Realtime* dan aplikasi ini dibuat dengan bahasa pemrograman pascal dan *Borland Delphi 7.0* sebagai software development.

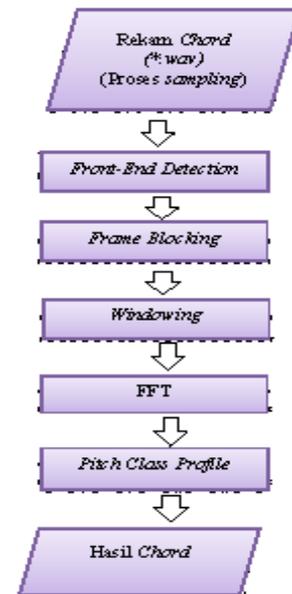
2. Perancangan Sistem

2.1 Gambaran Umum

Perancangan sistem merupakan tahap yang penting dalam proses pembuatan program. Perancangan bertujuan agar

dalam pembuatannya dapat berjalan secara sistematis, terstruktur dan rapi sehingga hasil program dapat berjalan sesuai dengan apa yang dikehendaki. Pembuatan sistem ini terdiri dari proses perekaman chord, front End Detection, Frame Blocking, Windowing dan yang terakhir PCP (*Pitch Class Profile*)

Secara umum pembuatan program ini mengikuti alur sesuai yang ditunjukkan pada gambar 1 berikut.



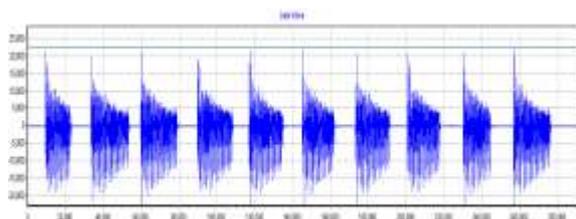
Gambar 1. Diagram perancangan system

2.1.1 Perekaman Chord

Input dari sistem yang dibuat adalah berupa file audio berformat wav. File ini didapat dari perekaman instrumen gitar dalam hal ini gitar bass dan gitar akustik. Setiap *chord* direkam dengan menggunakan instrumen gitar, yaitu sebanyak 10 sample. Format perekaman ini disimpan dalam format wav standar atau kualitas CD, yaitu 44100 Hz, 16 bit, stereo. Untuk merekam *chord* dari instrumen gitar tersebut, dibuat aplikasi record dengan memanfaatkan komponen waveaudio pada Delphi 7. Pada perekaman *chord* ini terjadi proses *sampling* yang mana sinyal analog sudah diubah menjadi sinyal digital.

2.1.2 Proses Pembacaan Sampling

Proses ini bertujuan untuk membaca data dari *file wave* yang kemudian disimpan dalam suatu array agar nantinya data dari *file wave* dapat diproses. Hal yang perlu dilakukan adalah membuat tipe data dasar untuk mengambil informasi yang ada pada header *file wave*[12].



Gambar 2. Data wave yang dibaca dan ditampilkan dalam chart

2.1.3 Front-End Detection

Tahap pertama proses pengolahan sinyal adalah *Front-End Detection*. Seperti namanya, *Front-End Detection* digunakan untuk menentukan batasan suatu sinyal dalam hal ini letak sinyal awal dan akhir dari suatu frame sehingga bentuk sinyal asli tidak berubah. Untuk menentukan letak awal dan akhir suatu sinyal dapat dilihat berdasarkan power sinyal. Rumus Power Sinyal adalah sebagai berikut :

$$P = \sum_{i=0}^{i=N} |X_i|^2 \dots\dots\dots (1)$$

Pada implementasi pembuatan sistem ini, *Front-End Detection* tidak hanya digunakan untuk mendeteksi awal dan akhir dari sinyal input saja. Namun digunakan untuk mendeteksi ada tidaknya setiap petikan gitar pada sinyal secara keseluruhan. Karena setiap petikan gitar yang direkam akan menghasilkan nilai data wav yang besar dan nilainya akan turun secara komulatif selama tidak terdapat petikan lagi.

2.1.4. Frame Blocking

Pada pembuatan sistem, proses *Frame Blocking* dijalankan pada proses *Front-End Detection*, karena perhitungan power sinyal pada *Front-End Detection* dilakukan dalam setiap *frame*. Pengambilan sample setiap *frame* diambil dalam waktu millisecond (ms). Berikut ini rumus yang digunakan untuk pengambilan sample tiap *frame* :

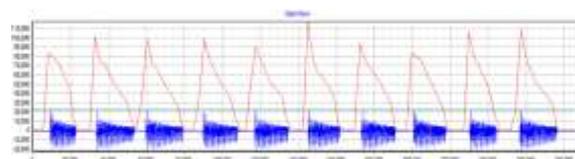
$$SFr = Fs * (t / 1000) \dots\dots\dots(2)$$

Keterangan :

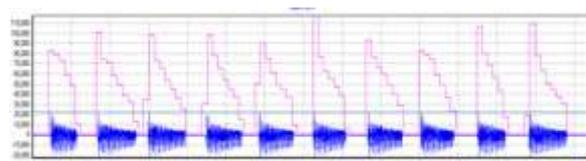
- SFr : Sample per *Frame*
- Fs : Frekuensi sample/Sample Rate
- t : Waktu pengambilan (ms)

Nilai 1000 pada rumus di dapat dari satuan second menjadi milisecond, dalam hal ini 1 s = 1000 ms. Nilai sample per *frame* tergantung dari frekuensi sample dan waktu pengambilan. Jika nilai frekuensi sample sebesar 44100Hz, maka hal ini berarti setiap 1 detik terdapat 44100 sample. Dalam pembuatan sistem digunakan waktu. 62 ms. Maka nilai sample per *framenya* sebesar 2734.

Di bawah ini hasil frond end dan frame blocking:



(a)



(b)

Gambar 3. Tampilan (a) *Front end Detection*
(b) *Frame Blocking*

2.1.5 Windowing

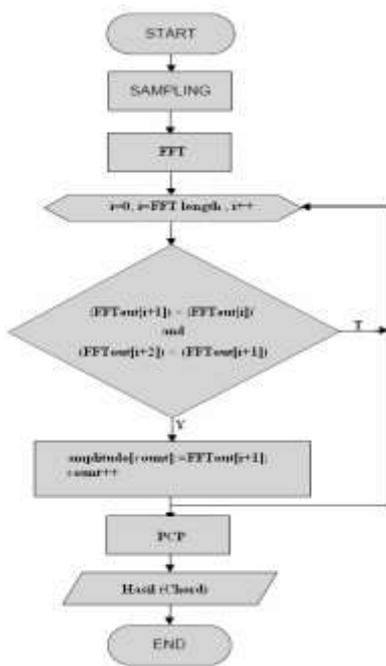
Proses windowing ini dijalankan pada proses FFT, karena fungsi windowing ini jadi satu dalam tool FFT pada komponen DSPlab. Jadi hanya memanggil fungsi windowing yang disediakan oleh komponen tersebut.

2.1.6 Fast Fourier Transform

Pada implementasi pembuatan sistem, nilai awal sinyal ini akan menjadi parameter pada fungsi FFT. Untuk prosesnya, mulai titik awal sinyal akan diambil sample sebanyak *buffesize* yaitu 16384 dari FFT yang digunakan untuk dimasukkan dalam variable array RealIn yang selanjutnya akan dihitung magnitudenya.

2.1.7 Deteksi Puncak

Sebelum masuk ke proses pitch class Profile maka dilakukan pendeteksian puncaknya terlebih dahulu. Setiap nilai puncak frekuensi tersebut akan dideteksi nilai puncak tertingginya untuk dapat dimasukkan ke dalam 12 nada berdasarkan nilai satuan frekuensi nada.



Gambar 4. Flowchart Deteksi Puncak

2.1.8 Pitch Class Profile

Setelah diperoleh nilai puncak frekuensi, maka nilai tersebut dikelompokkan berdasarkan satuan frekuensi nada. Dalam mengelompokkan ke dalam frekuensi nada, diperlukan perhitungan jarak awal dan akhir disetiap nada, sebagai range dari nada tersebut. Berikut ini rumus range nada yang dipakai :

$$\text{Awal nada ke-}i = (f. \text{nada ke-}(i-1) + f. \text{nada ke-}i)/2 \quad (3a)$$

$$\text{Akhir nada ke-}i = (f. \text{nada ke-}i + f. \text{nada ke-}(i+1))/2 \quad (3b)$$

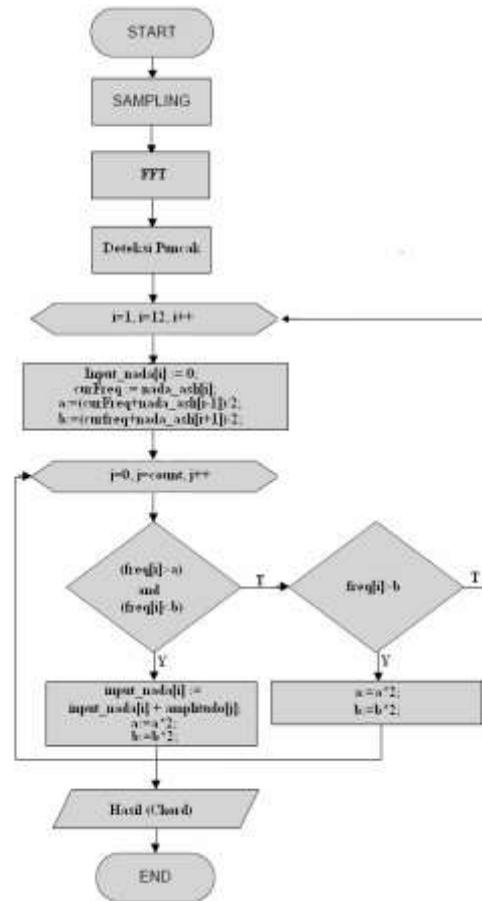
Contoh untuk nada A :

$$\begin{aligned} \text{Awal} &= (\text{frek. nada G\#} + \text{frek. nada A})/2 \\ &= (103.83 + 110.00)/2 \\ &= 213.83/2 \\ &= 106.92 \end{aligned}$$

$$\begin{aligned} \text{Akhir} &= (\text{frek. nada A} + \text{frek. nada A\#})/2 \\ &= (110.00 + 116.54)/2 \\ &= 113.27 \end{aligned}$$

Rumus diatas digunakan untuk semua nada. Setelah semua nada memiliki nilai range, maka setiap nilai puncak frekuensi akan dikelompokkan berdasarkan range-range nada.

Berikut ini flowchart dari proses *Pitch Class Profile* hasil perhitungan *Pitch Class Profile* untuk semua nada :



Gambar 5. Flowchart Deteksi Puncak

2.1.9 Mean Pitch Class Profile

Mean Pitch Class Profile ini merupakan rata-rata nilai PCP yang dihitung dari semua sinyal yang terdeteksi. Mean PCP digunakan untuk mengurangi error pada hasil akhir sistem yang mana chord yang terbanyak yang diambil untuk hasil akhir dari sistemnya.

3. Pengujian dan Analisis Sistem

Pengujian aplikasi terlebih dahulu dilakukan dengan mengambil data uji berupa file wave yang akan direkam dengan menggunakan aplikasi perekaman yang ada dalam aplikasi :

3.1 Perekaman File Wave

Langkah awal dalam pengujian ini adalah perekaman chord menggunakan instrumen-instrumen tunggal berupa gitar akustik dan gitar bass dengan menggunakan aplikasi record yang telah dibuat. Dalam perekaman ini dilakukan perekaman 10 sample disetiap chord karena setiap petikan nada akan menghasilkan hasil FFT yang berbeda. Hal ini disebabkan dari banyak faktor, khususnya saat proses perekaman. Hardware sound card menentukan kualitas

baiktidaknya suatu hasil perekaman. Sistem pengkabelan juga dapat mempengaruhi hasil perekaman, karena pada sistem pengkabelan yang kurang baik dapat menghasilkan noise yang dapat mempengaruhi hasil akhir. Noise juga dapat dihasilkan dari kualitas pick up gitar yang digunakan.

Gitar adalah instrumen string yang harus dituning sebelum memainkannya. Kurang tepatnya pada saat tuning sangat mempengaruhi hasil frekuensi, karena tuning inilah yang menentukan benar tidaknya suatu frekuensi.. Dari segi pemain juga berpengaruh, yaitu cara menekan dan memetik gitar pada saat perekaman. Dalam penelitian ini chord yang digunakan hanya chord Major saja. Jadi perekaman file wav terdapat 24 macam, yakni 12 file wav dari gitar akustik dan 12 file wav dari gitar bass.



Gambar 6. Tampilan perekaman pada aplikasi

3.2 Pengujian Aplikasi

Setelah mendapatkan file wavnya. Maka file tersebut akan dimasukkan ke dalam aplikasi. Di bawah ini merupakan tampilan awal dari aplikasi.

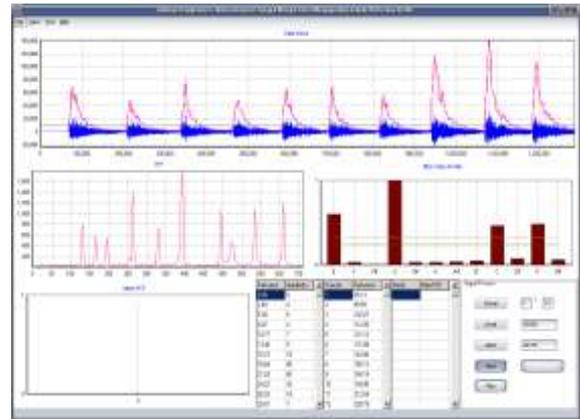


Gambar 7. Tampilan form utama aplikasi

Untuk membuka file wave yang sudah direkam, maka klik open pada menu file. Setelah open file dilakukan, pada form utama akan menampilkan hasil pembacaan file audio, FFT pada sinyal pertama yang ter Front-End Detection, nilai PCP dan tabel-tabel nilai FFT dan deteksi puncak.

Pada pengujian aplikasi ini chord yang di uji adalah chord C Major yang telah direkam dengan alat musik gitar akustik yaitu sebanyak 10 petikan gitar. Setelah chord

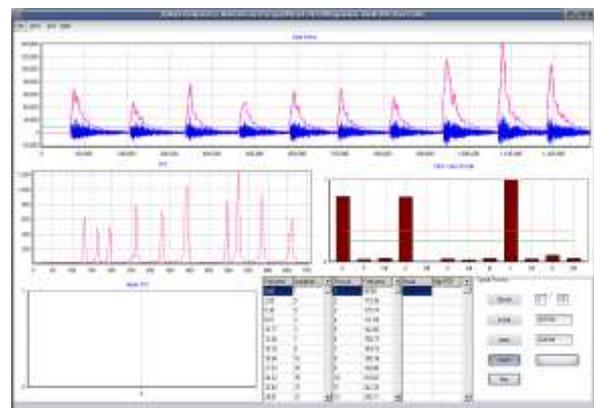
dimasukkan kemudian aplikasi akan memproses file wav tersebut. Untuk proses awal seperti yang telah dijelaskan sebelumnya bahwa aplikasi ini akan mendeteksi setiap petikan senar gitar yang direkam kemudian sinyal tersebut dibagi tiap frame sebanyak 2734 sample per frame. Hingga hasil akhir setelah file wav dimasukkan maka aplikasi akan menampilkan gambar seperti pada gambar 8 berikut ini:



Gambar 8. Tampilan setelah open file, hasil FFT dan PCP sample ke-1 chord C Major

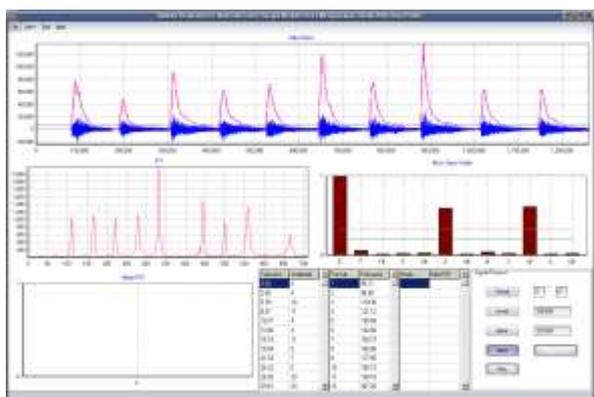
Berdasarkan gambar di atas ada beberapa sinyal yang dideteksi oleh sistem yaitu sebanyak 13 sinyal. Dari data masukan yang diambil sebanyak 10 sample terdeteksi 13 sinyal hal ini dikarenakan ada beberapa sinyal data yang setelah di power melebihi batas standar deviasi sehingga terdeteksi suatu petikan gitar.

Pada gambar tersebut juga, sinyal pertama memiliki nilai awal sebesar 68350 dan nilai akhir 84735. Nilai ini menunjukkan awal dan akhir dari satu sinyal yang mana satu sinyal ini menunjukkan satu suara petikan gitar. Untuk mengetahui nilai sinyal selanjutnya klik tombol next seperti yang terlihat pada gambar 9:



Gambar 9. Hasil FFT dan PCP sample ke-2 sinyal chord C Major

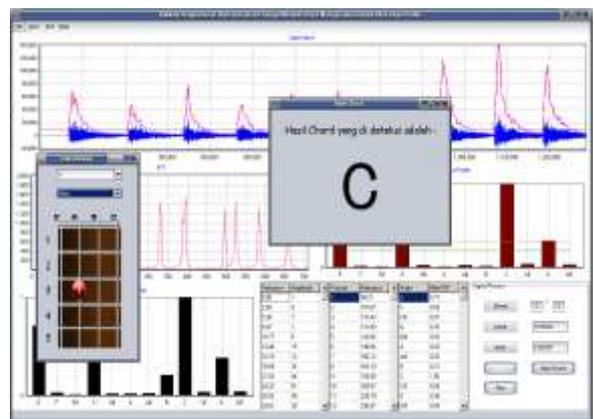
Gambar 9 merupakan hasil setelah tombol next diklik. Jika dibandingkan antara gambar 9 dengan gambar 8 terjadi perubahan posisi pada 2 chartnya yaitu pada chart FFT dan PCP. Hal ini dikarenakan sistem mendeteksi sinyal atau petikan selanjutnya yang mana dari ke 13 sinyal atau petikan gitar tersebut memiliki perbedaan gelombang antara sinyal satu dengan lainnya. Gambar untuk sinyal selanjutnya dapat dilihat pada gambar 10 :



Gambar 10. Hasil FFT dan PCP sample ke-3 sinyal chord C Major

Dari gambar 8 sampai gambar 10 diatas adalah sebagian hasil FFT dan PCP dari chord C Major. Jadi dapat diketahui bahwa hasil FFT dan PCP antara sample 1 dengan yang lain berbeda, penyebab ini seperti yang dijelaskan pada proses perekaman sebelumnya.

Setelah semua sinyal terproses hingga sinyal akhir yaitu sinyal ke 13, maka nilai rata-rata PCP akan keluar bersamaan dengan hasil chord dan chord dialognya. Hasil akhir dari sistem didapat bahwa chord yang dideteksi adalah chord C Major seperti gambar di bawah ini :



Gambar 11. Hasil akhir sistem

Jadi dapat disimpulkan bahwa hasil perekaman chord C Major sama dengan hasil akhir dari sistem yang terdiri dari 3 nada dasar yang dominan, yaitu C, E, G sesuai dengan teori chord yang terdapat pada bab II. Hal itu

dikarenakan suara dengungan senar gitar sebelum mencapai titik nada yang benar. Untuk melihat lagi hasil akhir mengenai chord apa yang dideteksi klik pada button Hasil Chord. Tombol ini akan aktif pada saat semua sinyal telah terproses.

Untuk mengetahui pengenalan chord yang lain kita dapat mengulangi langkah-langkahnya dari awal sesuai yang telah dijelaskan sebelumnya. Dari proses perekaman hingga hasil akhir yang dilakukan sebelumnya, dan di bawah ini merupakan tabel hasil uji aplikasi dalam mendeteksi chord lainnya berdasarkan hasil perekaman yang telah dilakukan.

Tabel 1. Pengenalan Chord Major Alat Musik Gitar Akustik

File Wav	Diuji	Benar	Error	Hasil Akhir Chord	Persent (%)
A	10	9	1	A	90
A#	10	10	0	A#	100
B	10	10	0	B	100
C	10	10	0	C	100
C#	10	10	0	C#	100
D	10	10	0	D	100
D#	10	10	0	D#	100
E	10	10	0	E	100
F	10	10	0	F	100
F#	10	10	0	F#	100
G	10	10	0	G	100
G#	10	10	0	G#	100

Terlihat pada tabel bahwa aplikasi berjalan dengan baik dengan melihat persentase keberhasilan yang diperoleh pada setiap chordnya yang walaupun pada beberapa chord masih memiliki error. Error disini merupakan nilai PCP yang bernilai satu pada chord lain yang tidak sesuai dengan chord yang dimasukkan. Dari tabel 1 di atas chord yang memiliki persentase keberhasilan tertinggi adalah chord D# dengan perolehan persentase 100 %. Sedangkan tingkat pengenalan chord dengan nilai persentase terendah adalah chord D dengan perolehan persentase sebesar 18.75 %. Hasil ini dikarenakan pada saat perekaman file wave audio masih ada interferensi. Jadi, dengan melihat hasil pada tabel 1 tersebut maka didapat rata-rata persentase keberhasilan dari alat musik gitar akustik adalah 72.27 % dengan persentase error sebesar 27.72 % dihitung berdasarkan keseluruhan hasil persentase tiap chord. Sedangkan pada tabel 2 merupakan tabel pengenalan chord untuk alat musik bass. Pada pengujian dengan gitar bass dilakukan sama seperti pengujian rekaman file wav dengan gitar akustik. Tabel 2 ini merupakan tabel hasil pengenalan chord dari alat musik bass dari setiap nada yang sudah direkam:

Tabel 2. Pengenalan chord Major alat musik gitar bass

File Wav	Diuji	Benar	Error	Hasil Akhir Chord	Persent (%)
----------	-------	-------	-------	-------------------	-------------

A	11	7	4	A	63.6
A#	12	7	5	A#	58.3
B	10	9	1	B	90
C	13	10	3	C	76.9
C#	10	9	1	C#	90
D	16	3	13	A	18.75
D#	10	10	0	D#	100
E	14	10	4	E	71.4
F	10	7	3	F	70
F#	10	8	2	F#	80
G	15	11	4	G	73.3
G#	12	9	3	G#	75

Terlihat pada tabel bahwa aplikasi pengenalan *chord* menggunakan alat musik gitar bass berjalan dengan baik dengan melihat persentase keberhasilan dari setiap hasil chordnya. Jadi berdasarkan tabel 2 tersebut alat musik gitar bass memiliki rata-rata persentase keberhasilan sebesar 99.16 % dengan persentase *error* sebesar 0.84 % yang terjadi pada *chord* A. Jika dibandingkan antara hasil akhir gitar akustik dan bass, gitar bass memiliki hasil yang baik, hasil ini diperoleh dikarenakan proses perekaman yang baik dan juga dikarenakan cara memainkan gitar bass yang hanya memerlukan satu jari untuk menekan satu “*key*” di tiap fretnya sehingga memungkinkan tidak terjadinya *error* di tiap file wav nya.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis dari aplikasi, tingkat pengenalan untuk masing-masing chord yang direkam menggunakan gitar akustik adalah 72.27 % dan gitar bass yaitu sebesar 99.16 %.

Kemudian dari hasil pengujian data wav sesuai file wave yang direkam, tingkat pengenalan tertinggi pada gitar akustik adalah pada saat sistem mengenali chord D#, dengan tingkat pengenalan sebesar 100 %. Sedangkan tingkat pengenalan terendah adalah pada saat aplikasi mengenali chord D yang mana memiliki persentase keberhasilan sebesar 18.75 %.

Untuk data uji menggunakan gitar bass, hanya satu chord saja yang mengalami error yaitu chord A dengan persentase keberhasilan sebesar 90 %. Sedangkan tingkat pengenalan tertinggi adalah pada semua chord selain chord A yaitu sebesar 100 %.

Oleh karenanya, dapat dikatakan bahwa semakin baik kualitas perekaman maka semakin baik hasil akhir dari sistem, hal ini ditandai dengan hasil uji aplikasi pada saat uji coba menggunakan instrumen tunggal gitar bass dengan perolehan rata-rata persentase keberhasilan sebesar 99.16 %. Sebaliknya, semakin kurang kualitas perekaman maka semakin kurang baik hasil akhir dari sistem yang didapatkan, hal ini ditandai dengan tingkat persentase yang diperoleh pada saat uji coba menggunakan gitar akustik yaitu dengan perolehan persentase error sebesar 27.72 %.

Mengingat masih banyaknya hal-hal yang belum dapat diimplementasikan pada penelitian ini, maka penulis mempertimbangkan beberapa saran untuk perbaikan-perbaikan penelitian ini dalam hal fitur atau alat untuk mengurangi noise pada perekaman audio seperti mixer agar mendapatkan hasil perekaman yang baik. Penelitian ini dapat dikembangkan lebih lanjut menjadi aplikasi pendeteksian *chord* pada suatu lagu.

Daftar Pustaka

- [1]. Junaidy Permana, Febrianzah, Pembuatan Database Software Musik Digital Mentor, Skripsi D-3 Politeknik Elektronika Negeri Surabaya : 2009
- [2]. Fitriani, Mila, Pembuatan Database Transkrip Akord Instrumen Tunggal Menggunakan Metode Enhanced Pitch Class Profile (Epcp), Skripsi D-3, Politeknik Elektronika Negeri Surabaya , 2010.
- [3]. Realtime Chord Recognition of Musical Sound : a System Using Common Lisp Music, Takuya Fujishima, CCRMA, Stanford University.
- [4]. [4] Akbar, Fandy, Pembuatan Software Musik Digital Mentor”, Skripsi D-3, Politeknik Elektronika Negeri Surabaya, 2009
- [5]. [5] Wahana Komputer (2002).Panduan Praktis Pemograman Delphi 7. Yogyakarta: Penerbit Andi Offset.
- [6]. Gold, Ben, Morgan, Nelson.(2000). Speech and Audio Signal Processing. New York: John Wiley & Sons, Inc.
- [7]. Sumi Kouhei, Katsutoshi Itoyama, Kazuyoshi Yoshii, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, “Automatic Chord Recognition Based On Probabilistic Integration Of Chord Transition And Bass Pitch Estimation”, Dept. of Intelligence Science and Technology National Institute of Advanced Industrial Graduate School of Informatics, Kyoto University Science and Technology (AIST) Japan,2008
- [8]. ---, Audio Sampling, <http://www.elektroindonesia.com/elek35a.html>, Agustus 2012
- [9]. ---, Pengenalan Chord, <http://ratidix.wordpress.com>, Agustus 2012.
- [10]. ---, *Front-End Detection*, <http://agusslamet.wordpress.com/2008/09/front-end-detection>, Agustus 2012.
- [11]. ---, *Wave*, <http://en.wikipedia.org/wiki/wav>, Agustus 2012.
- [12]. ---, *Akord*, <http://id.wikipedia.org/wiki/Akord>, Agustus 2012.
- [13]. ---, *Load & Display WAVE File (.WAV)*, <http://pebbie.wordpress.com/2007/08/08/load-display-wavefile-av.2012>.
- [14]. ---, *Kelebihan Pemrosesan Sinyal Digital*, <http://agfi.staff.ugm.ac.id/blog/index.php/2008/11/kelebihan-pemrosesan-sinyal-digital/>, Agustus 2012.
- [15]. ---, *Pengolahan Sinyal Digital*, <http://adhit8.blogspot.com/2010/04/pengolahan-sinyal-digital-menggunakan.html>, Agustus 2012
- [16]. ---, *Window Function*, http://en.wikipedia.org/wiki/Window_function, Juni 2012.
- [17]. ---, *Musik Tutorial (part-1) – Tangga Nada*, <http://Silentman13.wordpress.cm> Agustus 2012
- [18]. ---, *Fast Fourier Transform*, http://en.wikipedia.org/wiki/Fast_Fourier_transform, Agustus 2012
- [19]. ---, *Teori Recording*, <http://mellykustina.blogspot.com/2009/03/teori-recording.html>, Agustus 2012.
- [20]. ---, *Komponen DSPLab*, <http://www.teworks.com>, Juli 2012.