

PERANCANGAN APLIKASI PENGENALAN CHORD INSTRUMEN TUNGGAL MENGGUNAKAN TRANSFORMASI WAVELET DAN KEY DETECTION

Andi Pangerang^{*)}, Achmad Hidayatno, and Ajub Ajulian Zahra

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

^{*)}Email: pangerang_andi@yahoo.com

Abstrak

Chord atau akor adalah susunan tiga nada atau lebih yang dibunyikan serempak sehingga menghasilkan suara yang harmonis. Kepekaan pendengaran manusia dan pengetahuan musik tidak dimiliki semua orang khususnya pemula, sehingga diperlukan sistem yang dapat mengenali chord. Pengenalan chord adalah bentuk transkripsi masukan file audio menjadi chord yang dapat mengiringi lagu. Tujuan penelitian ini adalah membuat perangkat lunak yang dapat mendeteksi chord instrument tunggal dengan keluaran berupa histogram Pitch Class Profile (PCP). Tahapan pertama adalah perekaman suara, kemudian prapengolahan: pengkonversian format stereo-to-mono, crop data dan normalisasi data. Selanjutnya file yang sudah dikonversi ke bentuk digital diekstraksi menggunakan transformasi Wavelet. Selanjutnya dilakukan key detection: mengubah spektrum dekomposisi paket wavelet menjadi spektrogram, chromagram dan histogram PCP. Apabila ada salah satu pitch yang sama dengan pitch suatu chord dari tiga pitch tertinggi yang terdeteksi, maka chord dikelompokkan sesuai dengan chord yang dibandingkan. Penelitian ini dibuat menggunakan bahasa pemrograman Matlab dengan masukan 120 sinyal chord mayor dan minor dari 12 pitch berbeda. Ukuran frame yang digunakan: 256, 512, 1024, 2048, dan 4096 sampel. Hasil pengujian menunjukkan dengan parameter: tingkat dekomposisi kesembilan, indukan wavelet Symlet, dan ukuran frame 1024 sampel, tingkat keberhasilan rata-rata sistem mengenali chord mayor dan minor berturut-turut adalah 92,67% dan 94,1%.

Kata Kunci : Chord, Instrumen Tunggal, Wavelet, Key Detection

Abstract

Chord or Accord is arrangement of three or more notes played simultaneously, thus produce harmonic sound. Good human hearing sensitivity and musical knowledge is not common especially beginners. Therefore, it is necessary to create a system which is detecting chord well. Chord recognition is audio input transcription from audio file into chord used to accompany the song. The purpose of research is to make a software to recognize single instrument chord producing Pitch Class Profile (PCP) histogram. First step is sound recording, preprocessing i.e. stereo-to-mono conversion, data cropping and data normalization. Afterward, digital file would be extracted using Wavelet Transformation. Later, key detection process: change wavelet packet decomposition spectrum to spectrogram, chromagram and PCP histogram. If there is one of highest three pitches detected similar with a pitch of chord, therefore chord is classified by comparing chord. research based on Matlab language programming and 120 major and minor chord signal from 12 various pitches as input. It uses five frame sizes i.e. 256, 512, 1024, and 4096 samples per frame. Test result showed, major and minor chord recognition success rate is 92,67% and 94,1% with parameters: ninth decomposition level, Motherwavelet Symlet and frame size 1024 samples.

Keywords : Chord, Single Instrument, Wavelet, Key Detection

1. Pendahuluan

Dalam kehidupan sehari-hari, manusia tidak bisa lepas dari yang namanya musik. Dalam dunia musik, terdapat istilah Chord atau akor yang biasa disebut sebagai "Kunci". Chord adalah gabungan beberapa nada yang

dibunyikan sehingga menciptakan suara harmonis. Dalam sebuah lagu, chord dapat dimainkan atau dibunyikan secara terpisah-pisah maupun secara bersamaan. Secara garis besar, chord dikategorikan menjadi beberapa tipe, antara lain: chord mayor, minor, *diminished*, *augmented*, *suspended*, *seventh* (7th), dan lain-lain. Dalam

memainkan sebuah lagu, pemain musik harus mengerti dan mencari chord-chord apa saja yang ada pada 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 memahami dalam mengenali karakteristik chord tertentu.

Chord Recognition atau pengenalan chord adalah bentuk transkripsi atau konversi dari suara berupa file audio menjadi chord, yang mana dapat dikelompokkan sesuai dengan karakteristik masing-masing mengingat ada berbagai jenis chord seperti mayor, minor, *augmented*, *diminished*, dan sebagainya. Fungsi untuk mengenali chord secara otomatis sangat penting dalam berbagai aplikasi musik seperti sistem musik interaktif, aplikasi informasi musik berdasarkan konten atau pencarian sampel dan lagu tematik dalam basis data audio, serta aplikasi edukatif bagi para pemula yang ingin mempelajari chord.^[2]

Pada perancangan penelitian ini akan dirancang sebuah perangkat lunak yang mampu mengenali chord secara otomatis apabila diberi masukan file audio dengan keluaran berupa huruf dan dialog kunci serta informasi tentang jenis chord beserta diagramnya, sehingga dapat membantu para pemula dalam mempelajari musik maupun mengetahui karakteristik setiap chord dalam sebuah lagu. Tujuan dari penelitian ini adalah untuk merancang perangkat lunak yang mampu mengenali chord secara otomatis dari instrumen tunggal berupa masukan file audio dengan proses pengolahan sinyal audio sehingga diperoleh keluaran yang dapat dikenali oleh sistem berupa informasi chord yang dapat dimanfaatkan oleh pemula untuk membantu mempelajari dan memahami suatu chord.

Dalam penelitian ini penulis membatasi permasalahan sebagai berikut:

1. File input yang digunakan adalah file audio dengan format MP3 dan WAV.
2. File yang direkam berupa instrumen tunggal
3. Hasil keluaran berupa informasi *chord* berupa huruf dan dialognya.
4. *Chord* yang dikenali adalah *chord* dasar dan substitusi.
5. Aplikasi ini bersifat *offline*
6. Aplikasi ini dibuat menggunakan perangkat lunak Matlab R2011a.

2. Metode

2.1 Teori Dasar Musik

Dalam seni musik, dikenal istilah tangga nada. Istilah tersebut berisikan kumpulan nada-nada yang harmonis. Nada-nada yang harmonis ini terjadi karena ada aturan dibalik itu semua. Di bawah ini akan dijelaskan mengenai

tangga nada, diantaranya tangga nada kromatik dan tangga nada diatonik, serta *chord*.

2.1.1 Tangga Nada Kromatik

Tangga nada kromatik adalah kumpulan dari semua nada dalam musik. Dalam hal ini, tangga nada kromatik berarti nada dari setiap “warna” atau menyatakan frekuensi yang berbeda-beda sama halnya seperti cahaya. Dikarenakan nada selalu berulang untuk tiap oktaf yang ada, maka istilah ‘tangga nada kromatik’ sering dipakai untuk ke-12 nada dari tiap oktaf.^[2] Tabel berikut ini menunjukkan frekuensi dari ke-12 nada antara nada A pada 440 Hz dan nada A satu oktaf di atasnya.

Tabel 1. Tangga Nada Kromatik

Tangga Nada Kromatik			
Oktaf ke-3	Frekuensi	Oktaf ke-4	Frekuensi
A3	220,00 Hz	A4	440,00 Hz
A#3	233,08 Hz	A#4	466,16 Hz
B3	246,94 Hz	B4	493,88 Hz
C3	261,63 Hz	C4	523,25 Hz
C#3	277,18 Hz	C#4	554,37 Hz
D3	293,66 Hz	D4	587,33 Hz
D#3	311,13 Hz	D#4	622,25 Hz
E3	329,63 Hz	E4	659,26 Hz
F3	349,20 Hz	F4	698,46 Hz
F#3	369,99 Hz	F#4	739,98 Hz
G3	392,00 Hz	G4	784,00 Hz
G#3	415,30 Hz	G#4	830,60 Hz

Perbedaan antara dua buah pitch (nada) yang berdekatan disebut sebagai semiton. Meskipun ada 12 nada dalam 1 oktaf, tapi hanya 7 huruf pertama dari abjad yang dipakai untuk memberi nama pada nada, yaitu dari A sampai G. Kelima nada yang lain dalam tangga nada kromatik diberi nama dengan menempatkan tanda kres (#) atau tanda mol (b) setelah notasi nada.

2.1.2 Tangga Nada Diatonis

Skala diatonik memiliki tujuh not yang berbeda dalam satu oktaf. Pada tangga nada diatonik di bagi dua yaitu :

a. Mayor scales (Tangga nada Mayor)

Skala mayor merupakan tangga nada yang mempunyai jarak antar nada dengan rumus interfal 1-1-½-1-1-1-½

b. Minor scales (Tangga nada Minor)

Skala minor merupakan tangga nada yang mempunyai jarak antar nada dengan interfal 1-½-1-1-½-1-1 (untuk tangga nada minor alami / melodis menurun).

Tangga nada minor sendiri dapat dibagi menjadi tiga:

1. Tangga Nada Minor Alami (melodis menurun):
Interval 1-½-1-1-1-1-½
2. Tangga Nada Minor Melodis (melodis menaik):
Interval 1-½-1-1-½-1-1

3. Tangga Nada Minor Harmonis:
Interval 1-1/2-1-1-1/2-1-1/2

2.1.3 Chord (Akor)

Chord atau akor adalah gabungan beberapa nada dasar yang dibunyikan secara serempak ataupun terpisah sehingga menghasilkan suara harmonis. Umumnya antara chord dan kunci adalah hal yang sama bagi kebanyakan orang awam, namun secara istilah kedua hal ini berbeda. Chord dalam istilah musik merujuk pada tiga nada dasar atau lebih yang dibunyikan baik secara serempak maupun terpisah. Sementara kunci atau *key* merujuk pada nada dasar yang dipakai dalam sebuah lagu. Istilah *key* dalam not balok ditandai dengan notasi kres (#) yang berarti nada dinaikkan setengah dari nada dasarnya dan mol (b) yang berarti nada diturunkan setengah nada dari nada dasarnya. Jenis-jenis chord adalah sebagai berikut:

a. Chord Mayor

Chord *Mayor* biasanya dituliskan hanya berupa huruf kapital seperti C, D, E, F, G, A, B. Mencari chord *mayor* dapat dilakukan dengan menggunakan nada ke 1 – 3 – 5 dari tangga nada C mayor, Contoh : *Chord C = Nada C – E – G*

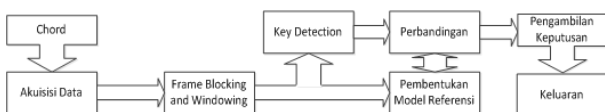
b. Chord Minor

Chord *Minor* biasanya dituliskan dengan penambahan karakter ‘m’ setelah huruf kapital seperti Cm, Dm, Em, Fm, Gm, Am, Bm. Ada pula yang menuliskan chord minor dengan huruf kecil saja seperti c, d, e, f, g, a, b. Apabila salah satu chord mayor sudah diketahui (misalnya C mayor), maka chord minornya (yaitu C minor) dapat dengan mudah ditentukan yaitu dengan cara menurunkan nada yang ada ditengah sebanyak setengah interval. Contoh : *Chord Cm = Nada C – D#/Eb – G*.

Apabila ditemukan nada maupun chord seperti D#/Eb, itu berarti bunyi dari D# terdengar sama dengan bunyi Eb. Inilah yang disebut sebagai chord atau nada *enharmonis*. Karena chord atau nada seperti ini memiliki nama berbeda namun bila dimainkan menghasilkan frekuensi dan suara yang sama.^[2]

2.2 Pengolahan Sinyal Digital

Sistem pengenalan chord memiliki diagram blok yang dapat dilihat pada Gambar 1 berikut ini:



Gambar 1. Diagram Blok Pengenalan Chord

2.2.1 Akuisisi Data Suara Digital

Sebagian besar sinyal-sinyal yang kita jumpai sehari-hari, adalah sinyal analog. Agar dapat memproses sinyal analog dengan alat digital, pertama-tama perlu mengkonversinya menjadi bentuk digital yaitu mengkonversi menjadi suatu deret angka yang mempunyai presisi nilai terbatas. Konversi A/D (*analog to digital*) melalui 3 tahapan proses yaitu (1). Proses sampling (2). Kuantisasi dan (3). Pengkodean.

Umumnya frekuensi sampling (f_s) yang digunakan untuk analisis sinyal suara berkisar pada rentang 6-20 kHz. Jika dilakukan proses sampling selama t detik dengan f_s adalah f ($f_s=f$) maka akan diperoleh suatu vektor V yang mempunyai elemen sebanyak $f \times t$ sebagai berikut : $V^T = [V_1 V_2 V_3 \dots V_n]$. Vektor V merepresentasikan sinyal digital. Selain itu, ada teori yang membahas tentang sampling yakni Teori *Nyquist-Shannon* yang menyatakan frekuensi sampling adalah paling sedikit dua kali frekuensi sinyal yang akan disampling sesuai rumus berikut.

$$f_s \geq 2f \quad (1)$$

Proses pengambilan sampel sinyal kerap kali terdapat derau yang menyebabkan perubahan bentuk sinyal. Agar dalam proses sampling bentuk sinyal aslinya tidak berubah dan terjadi *aliasing*, maka diperlukan *Front-End Detection* untuk menentukan batas awal dan akhir sinyal dari suatu *frame*. Sinyal dapat dilihat berdasarkan aras daya untuk menentukan letak awal dan akhir sinyal. Sinyal terlebih dahulu dibagi menjadi beberapa *frame* dan setiap *frame* memiliki aras daya tertentu. Berikut ini adalah perhitungan daya sinyal:

$$P = \sum_{i=0}^N |V_i|^2 \quad (2)$$

Nilai daya ini digunakan untuk membedakan sinyal tersebut merupakan sinyal suara (*voice*) atau bukan. Berdasarkan nilai daya diatas, didapatkan nilai rata-rata dan setelah menambahkan standar deviasi pada nilai tersebut, maka didapatkan nilai awal dan akhir dari suatu *frame*^[6].

2.2.2 Frame Blocking dan Windowing

Dalam interval waktu yang panjang, pola gelombang suara tidak stationer, tetapi dalam waktu yang cukup pendek (10-30 milidetik) dapat dikatakan stationer, dikarenakan kecepatan perubahan spektrum suara berkaitan dengan kecepatan perubahan organ-organ penghasil suara pada manusia yang dibatasi oleh keterbatasan fisiologi. Berdasarkan hal tersebut, sinyal suara digital yang telah diakuisisi dapat dibagi-bagi menjadi segmen-segmen digital berdurasi 10-30 milidetik yang disebut *frame* dan proses pembentukan *frame* disebut *frame blocking* dengan tiap *frame* direpresentasikan dalam sebuah vektor.^[6] Berikut ini

adalah rumus untuk mengetahui jumlah sampel setiap frame-nya:

$$SFr = f_s \times \frac{t}{1000} \quad (3)$$

Tahap selanjutnya dari pemrosesan sinyal adalah membuat *window* terhadap masing-masing frame dengan tujuan untuk meminimalkan ketidakkontinuan pada awal dan akhir setiap frame. Jika *window* didefinisikan sebagai $w(n)$, maka hasil dari penjendelaan sinyal adalah:

$$\tilde{x}_i(n) = \tilde{x}_i(n) \cdot \tilde{w}_i(n); 0 \leq n \leq N-1 \quad (4.a)$$

Umumnya, *window* yang digunakan adalah *window Hamming*. Pembentukan *window Hamming* menggunakan rumus sebagai berikut:

$$w(k+1) = 0,54 - 0,46 \cos\left(2\pi \frac{n}{N-1}\right); 0 \leq n \leq N-1 \quad (4.b)$$

Setelah frame blocking dan frame windowing selesai dilakukan maka pada tiap-tiap frame dilakukan ekstraksi ciri (*feature extraction*).

2.2.3 Ekstraksi Ciri (*Feature Extraction*)

Tujuan utama dari ekstraksi ciri adalah untuk mereduksi ukuran data tanpa merubah karakteristik dari sinyal suara dalam setiap framenya. Cara kerjanya adalah dengan mengkonversikan bentuk sinyal suara ke dalam bentuk representasi secara parameter.

2.2.3.1 Transformasi Wavelet Diskret

Transformasi adalah proses merepresentasikan suatu sinyal ke dalam kawasan lain. Tujuan dari transformasi adalah untuk lebih menonjolkan sifat atau karakteristik sinyal tersebut. Definisi wavelet (secara harfiah berarti “gelombang kecil”) adalah himpunan fungsi dalam ruang vektor L^2I , yang mempunyai sifat-sifat sebagai berikut:

- berenergi terbatas
- merupakan fungsi band-pass pada domain frekuensi
- merupakan hasil penggeseran (translasi) dan penskala (dilatasi) dari sebuah fungsi tunggal (induk), yaitu

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad (5)$$

dengan $a, b \in R$ (bilangan nyata), dan $a \neq 0$. Dalam hal ini a adalah parameter penskala dan b adalah parameter penggeser posisi terhadap sumbu t . Faktor normalisasi $\frac{1}{\sqrt{|a|}}$ digunakan untuk memastikan bahwa $|\psi_{a,b}(t)| = |\psi(t)|$.^[5]

Pada dasarnya, Transformasi Wavelet merupakan sebuah teknik pemrosesan sinyal multiresolusi. Keuntungan Transformasi Wavelet adalah bahwa jendelanya bervariasi. Untuk mengisolasi ketidakkontinuan sinyal, dapat digunakan fungsi basis yang sangat pendek. Pada

saat yang sama, untuk analisis frekuensi secara terperinci, dapat digunakan fungsi basis yang sangat panjang.^[4] Secara garis besar, Transformasi Wavelet dibedakan menjadi dua, yaitu Transformasi Wavelet Kontinu dan Transformasi Wavelet versi diskret. Versi diskret ada yang bersifat semi-diskret yang dikenal dengan runtun wavelet, dan ada yang diskret penuh yang dikenal dengan Transformasi Wavelet Diskret. Pada Transformasi Wavelet Kontinu, waktu t serta parameter penskala akan digeser berubah secara kontinu (dengan $a \neq 0$). Transformasi Wavelet kontinu fungsi $f(t)$ didefinisikan dengan:

$$TWK_j(a,b) = \langle \psi_{a,b}(t), f(t) \rangle \quad (6.a)$$

$$= \int_{-\infty}^{\infty} \psi_{a,b}(t) f(t) dt = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \psi_{a,b}(t) f(t) dt$$

Transformasi Wavelet Kontinu ini mempunyai dua kelemahan, yaitu reduksi dan ketidakpraktisan. Masalah tersebut dapat diselesaikan dengan mendiskretkan parameter a dan b .^[5] Pendiskretan a dan b menghasilkan Runtun Wavelet/RW, yaitu:

$$RW(a_0^j, a_0^j kb_0) = \frac{1}{\sqrt{a_0^j}} \int_{-\infty}^{\infty} \psi\left(\frac{t}{a_0^j} - kb_0\right) f(t) dt \quad (6.b)$$

dengan membuat waktu t menjadi diskret maka diperoleh Transformasi Wavelet Diskret (TWD), yaitu:

$$TWD(a_0^j, a_0^j kb_0) = \frac{1}{\sqrt{a_0^j}} \sum_n \psi\left(\frac{n}{a_0^j} - kb_0\right) f(n) \quad (6.c)$$

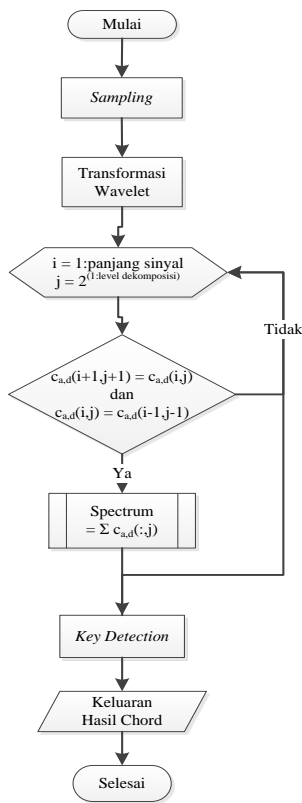
Untuk $a_0=2$ dan $b_0=1$, waveletnya disebut *dyadic*, yaitu:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t}{2^j} - k\right) \quad (6.d)$$

Sehingga, transformasi Wavelet diskretnya menjadi:

$$TWD_{dy}(2^j, 2^j k) = 2^{\frac{j}{2}} \sum_n \psi(2^j n - k) f(n) \quad (6.e)$$

Dalam wavelet *dyadic* ini, skala j disebut dengan *oktaf*.^[3] Sebuah teknik yang efisien untuk mengimplementasikan TWD adalah teknik analisis multiresolusi yang dikembangkan Mallat tahun 1988, yaitu Transformasi Wavelet Cepat yang diimplementasikan menggunakan *filter bank*. Sinyal masukan S dilewatkan melalui 2 tapis komplementer (low-pass H dan high-pass G), dan *downsampling* dengan membuang setiap data kedua, sehingga diperoleh koefisien aproksimasi c_A (komponen frekuensi rendah) dan koefisien detail c_D (komponen frekuensi tinggi). Proses ini dapat diiterasi dengan cara melanjutkan dekomposisi terhadap c_A . Dengan demikian suatu sinyal dapat didekomposisi menjadi komponen-komponen dengan resolusi yang lebih rendah.^[3] Berikut ini adalah diagram alir dalam proses deteksi frekuensi:



Gambar 2. Diagram Alir Proses Deteksi Frekuensi

2.2.4 Key Detection

Dalam key detection ini, penulis menggunakan *Pitch Class Profile* (PCP) pada lima belas detik pertama dan terakhir pada sebuah masukan yang dipakai untuk menghitung PCP rata-rata yang kemudian akan dibandingkan dengan model referensi. Jika jaraknya cukup dekat dengan kunci yang dibandingkan, berarti kunci tersebut adalah kunci dari suatu lagu. Dalam PCP terdapat vektor 12 nilai *real* yang menyatakan durasi kumulatif dari setiap kelas pitch dalam musik. Ada 12 kelas pitch yang digunakan untuk mendeteksi puncak dari hasil transformasi Wavelet yakni E, F, F#, G, G#, A, A#, B, C, C#, D dan D#. Kemudian data-data tersebut dimasukkan ke dalam range kelas nada yang sudah ditentukan.

Perhitungan PCP dapat dilihat pada persamaan dibawah ini:

$$f_{awal} = \frac{1}{2}(f_{i-1} + f_i), f_{akhir} = \frac{1}{2}(f_i + f_{i+1}) \quad (7)$$

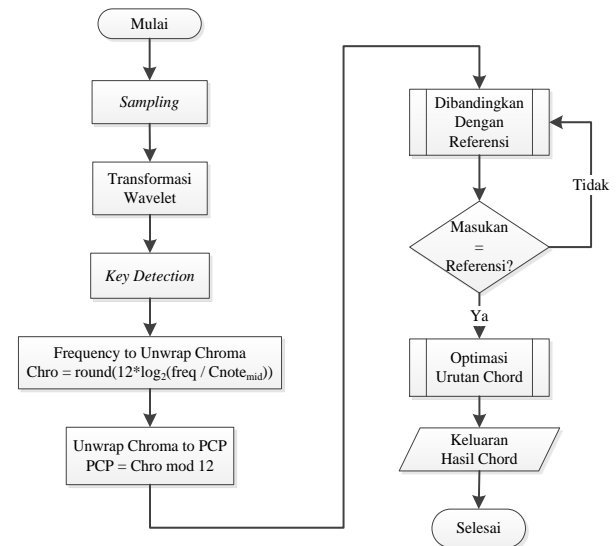
Berikut ini adalah tabel chord dengan nada dasar C Mayor yang diturunkan dari tangga nada mayor dan minor. Dapat dilihat pada tabel apabila nada dasar dari suatu lagu adalah C mayor, maka pitch *c, d, e, f, g, a, dan b* adalah pitch yang dominan sedangkan pitch lainnya seperti *c#* atau *g#* adalah pitch yang tidak memungkinkan untuk menghasilkan disonansi. Dari keseluruhan kunci mayor

dan minor yang berjumlah 24, hanya 10 kunci mayor dan minor yang digunakan pada nada dasar tertentu (kecuali fungsi VII [*leading note*] yang menggunakan chord diminished minor yang mirip dengan chord mayor 7 pada fungsi V [*dominant*]; Bdim: b – d – f ~ G7: g – b – d – f).^[1]

Tabel 2. Chord yang terpilih untuk kunci C Mayor

Pitch	c	d	E	f	G	A	b
Fungsi	I	II	III	IV	V	VI	VII
Chord yang Sesuai Skala	C	D _m	E _m	F	G	A _m	B _{dim}
Chord Sekunder Dominan	G	A	B	C	D	E	
Sekunder Subdominan	F	G	A	Bb	C	D	

Berikut ini adalah diagram alir dalam proses *Key-Detection*:



Gambar 3. Diagram Alir Proses Key Detection

2.2.5 Model Referensi PCP

Chord yang digunakan dalam pembuatan model referensi PCP adalah chord mayor dan minor. Dimasukkan pula beberapa sampel chord yang berbeda, dihitung rata-ratanya untuk setiap satu jenis chord dan dinormalisasi untuk kemudian disesuaikan secara manual agar memastikan nilai pitch dasarnya tidak berubah. Hal ini dilakukan untuk menghilangkan bias yang melekat pada salah satu atau beberapa chord lainnya karena hanya salah satu nada saja yang akan dianalisis PCP-nya. Contohnya, pitch untuk C mayor dan C minor tetap *c* dengan jarak yang sama dan pitch *c* inilah yang digunakan pada sistem untuk dibandingkan. Untuk setiap jenis chord pada suatu nada dasar, referensi PCP akan disimpan sesuai dengan nada dasarnya, sehingga agar mendapatkan duabelas chord dengan nada dasar yang berbeda diperoleh dengan menggeser satu nada untuk setiap *semitone*^[7].

2.2.6 Optimasi Urutan Chord (Pencocokan) dan Pengambilan Keputusan

Setelah tahapan key detection dan pembuatan model referensi PCP, hasil key detection akan dibandingkan dengan model referensi agar diperoleh chord sesuai dengan segala kemungkinan chord yang terjadi. Hal ini dilakukan sesuai dengan rumus kolerasi maksimum sebagai berikut^[1]:

$$key_x = \arg \max_x (cor(PCP_x, PCP_T^k) + 0,6 cor(Interval_x, Interval_T^k)) \quad (2.13)$$

Interval diperoleh dari dua pitch yang berbeda dengan rumus berikut:

$$Interval(x, y) = |num_x - num_y| \bmod 12 \quad (2.14)$$

Rumus diatas berlaku untuk data berupa digital. Sedangkan apabila masukannya berupa analog, estimasi interval sulit dilakukan secara presisi sehingga hanya diambil suku pertama dari persamaan berikut yakni:

$$key_x = \arg \max_x (cor(PCP_x, PCP_T^k)) \quad (2.15)$$

Dimana key_x adalah chord keluaran setelah dibandingkan, fungsi $cor(x,y)$ adalah fungsi korelasi, $argmax_x(x)$ adalah argumen maksimum, PCP_x adalah PCP hasil key deteksi dan PCP_T^k adalah PCP dari model referensi, $interval_x$ adalah interval hasil key deteksi dan $interval_T^k$ adalah interval T dari model referensi.

2.3. Implementasi Sistem

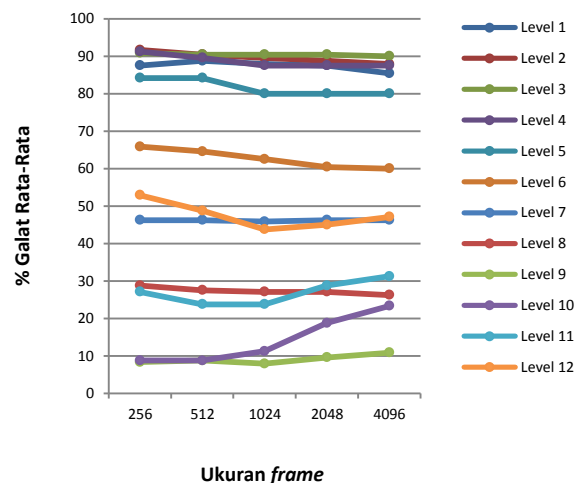
Pada penelitian ini sistem diimplementasikan dengan menggunakan GUI pada program Matlab R2011a, namun dalam pengimplementasiannya tidak akan lepas dari adanya dukungan perangkat keras. Perangkat keras yang digunakan meliputi: Laptop merk Axioo dengan spesifikasi sebagai berikut; *System Model* Axioo Neon HNM Model, *Processor* Intel (R) Celeron (R) CPU B820 ~1,70 GHz, *Memory* 2 GB RAM, Instrumen Keyboard, Kabel *Out*. Sedangkan perangkat lunak yang digunakan dalam PENELITIAN kali ini meliputi: *Operating system* yaitu Windows 7 Ultimate 32-bit, dan *Programing tools* berupa Audacity 2.0.3 dan Matlab R2011a

3. Hasil dan Analisa

Pada bab ini akan dibahas mengenai pengujian dan analisis sistem pengenalan chord pada instrumen tunggal dengan berbagai macam variasi yang digunakan. Kali ini tiga variasi digunakan untuk menguji dan menganalisis sistem apakah dapat mendeteksi *pitch* dan chord dengan benar, diantaranya yakni pengaruh ukuran *frame*, pengaruh penggunaan jenis indukan wavelet (*mother wavelet*) dan pengaruh tingkat dekomposisi sinyal tahapan ekstraksi ciri.

3.1. Pengaruh Tingkat Dekomposisi

Berdasarkan hasil pengujian yang diilustrasikan pada Gambar 4, diperoleh bahwa ketika menggunakan *frame* berukuran 256 dan 512, rataan persentase galat terkecil terjadi pada tingkat dekomposisi kesembilan, masing-masing 8,33% untuk *frame* berukuran 256 sampel dan 8,755% untuk *frame* berukuran 512 sampel. Sedangkan pada tingkat dekomposisi kesepuluh, rataan persentase galatnya sebesar 8,75% untuk *frame* berukuran 256 sampel maupun 512 sampel. Sementara itu, ketika menggunakan *frame* berukuran 1024, 2048 dan 4096, rataan persentase galat terkecil terjadi pada tingkat dekomposisi kesembilan, berturut-turut adalah 7,92% ; 9,58% ; dan 10,83%. Secara umum, rataan persentase galat terkecil terjadi pada tingkat dekomposisi kesembilan yakni sebesar 9,08%, sehingga dapat dikatakan bahwa sistem akan optimal dalam pendeteksian *pitch* ketika tingkat dekomposisi berada pada level kesembilan.



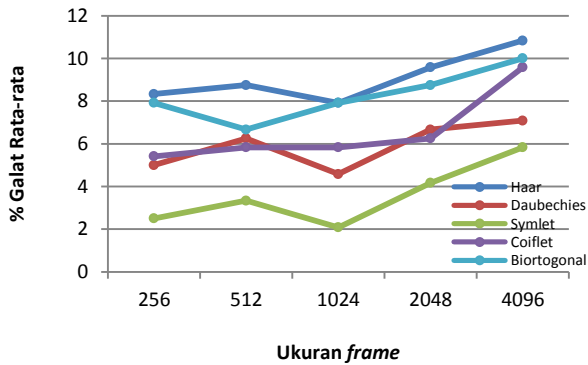
Gambar 4. Grafik Rataan Persentase Galat untuk Variasi Tingkat Dekomposisi

3.2. Pengaruh Indukan Wavelet

Setelah mengetahui pada tingkat dekomposisi ke berapakah posisi amplitudo tertinggi pada koefisien aproksimasi dan detail yang disusun menurut frekuensi dapat sesuai dengan frekuensi aslinya (jika menggunakan FFT), dari subbab sebelumnya diperoleh tingkat dekomposisi terbaik pada level ke-9, maka kali ini akan membandingkan jenis indukan wavelet (*mother wavelet*) yang dapat memengaruhi nilai *pitch*.

Berdasarkan hasil pengujian yang diilustrasikan oleh Gambar 5, ketika menggunakan *frame* berukuran 256, rataan persentase galat terkecil terjadi ketika menggunakan indukan wavelet *symlet*, yakni sebesar 2,5%. Ketika menggunakan *frame* berukuran 512, rataan persentase galat terkecil juga terjadi ketika menggunakan indukan wavelet *symlet*, dengan persentase sebesar

3,33%. Persentase galat terkecil untuk *frame* berukuran 1024, 2048 dan 4096 sampel berturut-turut adalah 2,08% ; 4,17% ; dan 5,83% dimana untuk ketiga *frame* tersebut persentase galat terkecil terjadi ketika menggunakan indukan wavelet *symlet*. Secara umum, rataan persentase galat terkecil terjadi ketika menggunakan indukan wavelet *symlet* yakni sebesar 3,58%, sehingga dapat dikatakan bahwa sistem akan optimal dalam pendeteksian *pitch* ketika menggunakan indukan wavelet *symlet*.



Gambar 5 Grafik Rataan Persentase Galat untuk Variasi Indukan Wavelet

3.3. Pengaruh Ukuran *Frame*

Selain penggunaan tingkat dekomposisi dan indukan wavelet, ukuran *frame* juga dapat memengaruhi frekuensi *pitch* yang dihasilkan. Setelah mengetahui pada tingkat dekomposisi terbaik dan indukan wavelet terbaik, ukuran *frame* akan dianalisis manakah yang mendekati *pitch* sebenarnya (jika dihitung menggunakan FFT). Pengujian kali ini akan menggunakan lima jenis *frame* mulai 256, 512, 1024, 2048 dan terakhir 4096. Sedangkan tingkat dekomposisi yang digunakan adalah tingkat dekomposisi terbaik pada level ke-9 dengan indukan wavelet terbaik yaitu jenis *Symlet*.

Berdasarkan tabel 3, rataan persentase galat terkecil terjadi ketika ukuran *frame* 1024 sampel, yang bernilai 2,08%. Sehingga dari tabel di atas dapat dikatakan bahwa ukuran *frame* terbaik yang digunakan agar pengenalan *pitch* dapat optimal adalah ketika *frame* berukuran 1024 sampel.

Tabel 3. Data hasil perhitungan persentase galat dan rataan persentase galat untuk setiap *frame*

Nama Chord	Persentase Galat untuk setiap <i>frame</i> (%)				
	256	512	1024	2048	4096
A mayor	0	10	0	10	0
A minor	10	0	0	10	10
A# mayor	0	0	0	0	10
A# minor	0	0	0	0	10
B mayor	0	0	0	0	0
B minor	0	0	0	0	0
C mayor	10	10	10	0	0

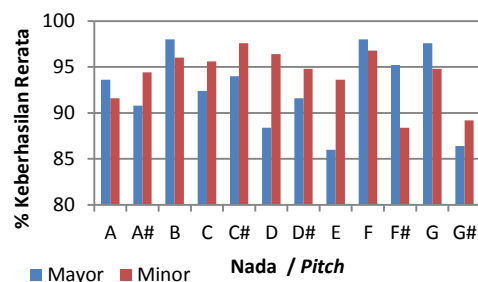
C minor	0	0	10	0	0
C# mayor	0	0	0	0	10
C# minor	0	0	0	0	0
D mayor	10	10	10	10	0
D minor	0	0	0	0	0
D# mayor	0	0	0	0	10
D# minor	0	0	0	10	20
E mayor	10	10	0	10	10
E minor	0	10	0	10	10
F mayor	0	0	0	0	10
F minor	0	0	0	10	0
F# mayor	0	0	0	0	10
F# minor	10	10	10	10	10
G mayor	0	0	0	0	0
G minor	0	0	0	0	0
G# mayor	10	10	10	10	10
G# minor	0	10	0	10	10
Rataan	2,5	3,33	2,08	4,17	5,83

3.4 Tingkat Pengenalan Chord

Berdasarkan hasil pengujian yang diilustrasikan oleh Gambar 6, tingkat keberhasilan rata-rata tertinggi untuk chord jenis mayor terjadi ketika mengenali chord B mayor dan F mayor dengan tingkat keberhasilan rata-rata sebesar 98%. Sedangkan tingkat keberhasilan rata-rata tertinggi untuk chord jenis minor terjadi ketika mengenali chord C# minor dengan tingkat keberhasilan rata-rata sebesar 97,6%.

Sedangkan tingkat keberhasilan rata-rata terendah untuk chord jenis mayor terjadi ketika mengenali chord E mayor dengan tingkat keberhasilan rata-rata sebesar 86%. Sedangkan tingkat keberhasilan rata-rata terendah untuk chord jenis minor terjadi ketika mengenali chord F# minor dengan tingkat keberhasilan rata-rata sebesar 88,4%.

Secara umum, tingkat keberhasilan sistem mengenali chord mayor adalah 92,67%, sedangkan tingkat keberhasilan sistem mengenali chord minor adalah 94,1%.



Gambar 6. Grafik Tingkat Keberhasilan Pengenalan Chord secara keseluruhan

4. Kesimpulan

Berdasarkan hasil pengujian 120 *chord* mayor dan minor dengan indukan wavelet *Haar*, tingkat dekomposisi terbaik untuk pendeteksian *pitch* adalah tingkat

dekomposisi kesembilan dengan rata-rata persentase galat keseluruhan untuk lima jenis ukuran *frame* (antara lain: 256, 512, 1024, 2048, dan 4096 sampel) sebesar 9,08%.

Pengujian dengan tingkat dekomposisi kesembilan, indukan wavelet (*mother wavelet*) terbaik untuk pendeteksian *pitch* adalah indukan wavelet *Symlet* dengan rata-rata persentase galat keseluruhan untuk lima jenis ukuran *frame* sebesar 3,58%.

Pengujian dengan tingkat dekomposisi kesembilan, jenis indukan wavelet *Symlet*, ukuran *frame* terbaik untuk pendeteksian *pitch* adalah ukuran *frame* 1024 sampel dengan rata-rata persentase galat sebesar 2,08%.

Pengujian dengan tingkat dekomposisi kesembilan, dengan variasi lima jenis indukan wavelet dan variasi lima jenis ukuran *frame*, rata-rata tingkat keberhasilan pengenalan chord tertinggi secara keseluruhan adalah 98% sedangkan rata-rata tingkat keberhasilan pengenalan chord terendah secara keseluruhan adalah 86%. Secara umum, tingkat keberhasilan sistem mengenali chord mayor adalah 92,67%, sedangkan tingkat keberhasilan sistem mengenali chord minor adalah 94,1%.

Perancangan ini dapat dikembangkan untuk dapat mengenali tidak hanya chord jenis mayor dan minor saja, melainkan chord jenis *diminished*, *augmented*, *suspended*, *dominant (seventh)* dan jenis chord lainnya secara *real-time* dan juga dapat dikembangkan untuk dapat mengenali sebuah lagu sederhana dengan urutan chord tertentu secara interaktif sehingga memudahkan seseorang yang ingin bermain musik tanpa harus mendalami pengetahuan tentang musik.

Referensi

- [1]. Aljanaki, Anna. (2011). *Automatic Musical Key Detection*. Master's Thesis, University of Tartu, Faculty of Mathematics and Computer Science, Institute of Computer Science.
- [2]. Gaffar, Imam. (2012). *Aplikasi Pengkonversi Nada-Nada Instrumen Tunggal menjadi Chord Menggunakan Metode Pitch Class Profile*, PENELITIAN S-1 Teknik Elektro Universitas Diponegoro.
- [3]. Krishnan, M., Neophytou, C.P., Prescott, G. (1994). *Wavelet Transform Speech Recognition Using Vector Quantization, Dynamic Time Warping and Artificial Neural Networks*, Telecommunication & Information Sciences Laboratory 2291 Irving Hill Drive, Lawrence, KS 66045.
- [4]. Mallat, Stephane. (1989), *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Pattern Anal. and Machine Intell., vol. 11, no. 7, pp 674-693.
- [5]. ————. (1999). *A Wavelet Tour of Signal Processing*, second edition, Academic Press 24-28 oval road, London NW1 7DX UK,
- [6]. Proakis J.G., Manolakis.D.G. (1997). *Pemrosesan Sinyal Digital*, edisi Bahasa Indonesia jilid 1, Jakarta: Prenhallindo
- [7]. Zenz, Veronica, Rauber, Andrean (t.th). *Automatic Chord Detection Incorporating Beat and Key Detection*, Vienna University of Technology Department of Software Technology and Interactive Systems Favoritenstrasse 9-11/188, A-1040 Vienna, Austria.