

**SISTEM CERDAS UNTUK MEMPREDIKSI *CHORDS* GITAR DARI LAGU MP3
DENGAN METODE *MEL-FREQUENCY CEPSTRAL COEFFICIENTS***

Rachmat Hidayat

Akademi Bina Sarana Informatika
Jl.Otista Raya No.25, Tangerang- Banten 15110

rachmat.rch@bsi.ac.id

Abstract

This study aims to identify and find the key tone of the song using ekstraksi cord mp3. An application that serves to extract the key songs on the guitar tune, so that will generate the key (Cord) tone that represents the humming. Technique Mel Frequency Cepstral Coefficients (MFCC), is the method used to extract features from music signals and comparing suaru with unknown speakers with the existing speakers in the database. This application input data undergo several processes, before finally producing the key (Cord) tone. These processes include identifying the sound of the song. Voice recognition is useful for biometric authentication and communication between the computer system.

Keywords: mel-frequency cepstral coefficients, vector quantization, K-means, tone, onset, humming.

1. PENDAHULUAN

Teknologi dan ilmu pengetahuan berkembang sangat cepat dan pesat juga mempengaruhi perkembangan seni musik, termasuk alat musiknya. Banyak sekali alat musik yang dihasilkan guna menunjang perkembangan seni musik. Gitar adalah salah satu alat musik yang mampu menyediakan nada yang jumlahnya tidak terbatas. Dalam hal bermain gitar, pengetahuan akan kunci gitar atau dikenal *Chord* sangat diperlukan. Bagi orang yang masih awam, mempelajari permainan gitar yang bersifat otodidak atau belajar sendiri tidak mudah karena Pemain gitar pemula harus mengetahui *Chord* gitar yang ada, untuk membuat sebuah *software* pendukung bermain gitar sehingga membantu *user* dalam mengenal dan mengingat kunci atau *Chord* gitar pada setiap lagu.

Sedangkan jumlah *Chord* gitar yang harus diingat cukup banyak dan bermacam-macam sehingga orang awam atau seorang pemain gitar pemula sangat sulit dalam mengingatnya. Penulis merancang dan membuat sebuah *software* pendukung bermain gitar sehingga membantu *user* dalam mengekstraksi kunci atau *Chord* gitar pada setiap lagu. *Software* ini akan melakukan proses analisa atau pengenalan teks lagu yang berisi suara dan akan di ekstrak kedalam teks dan proses ini antara lain mengidentifikasi suara lagu tersebut. Sistem ekstraksi berguna untuk otentikasi biometrik dan komunikasi antara komputer dengan sistem. Teknik *Mel Frequency Cepstral Coefficients* (MFCC) digunakan untuk ekstraksi ciri dari sinyal suaru musik dan membandingkan dengan penutur tak dikenal dengan penutur yang ada dalam *database*.

Filter bank digunakan sebagai pembungkus (*wrapping*) mel frekuensi. *Vector Quantization* (VQ) adalah proses meletakkan vektor-vektor ciri yang besar dan menghasilkan ukuran vektor-vektor yang kecil yang berhubungan dengan distribusi *centroid*. Algoritma *K-mean* digunakan untuk kluster. Dalam tahap pengenalan, ukuran distorsi berdasarkan minimisasi jarak Euclidean digunakan untuk mencocokkan penutur tak dikenal dengan penutur dalam *database*. *Database* suara musik menggunakan 10 penutur berbeda dengan MFCC 12, 20 *codebook*, dan 16 *centroid*.

2. LANDASAN TEORI

Pengertian Musik

Musik pada hakikatnya adalah bagian dari seni yang menggunakan bunyi sebagai media penciptaannya. Walaupun dari waktu ke waktu beraneka ragam bunyi, seperti klakson maupun mesin sepeda motor dan mobil, *handphone*, radio, televisi, *tape recorder*, dan sebagainya senantiasa mengerumuni kita, tidak semuanya dapat dianggap sebagai musik karena sebuah karya musik harus memenuhi syarat-syarat tertentu. Syarat-syarat tersebut merupakan suatu sistem yang ditopang oleh berbagai komponen seperti melodi, harmoni, ritme, timbre (warna suara), tempo, dinamika, dan bentuk. Sebelum lebih jauh membahas syarat-syarat tersebut berikut aspek-aspek lain yang terkait dengannya seperti sejarah musik, pencipta musik, karya-karya musik, dan berbagai formasi pertunjukan musik, bab ini akan terlebih dahulu meninjau beberapa definisi tentang musik, fungsi musik, dan jenis-jenis musik.

Pengertian Mp3

MPEG-1 Audio Layer 3 atau lebih dikenal sebagai MP3 adalah salah satu format berkas pengodean suara yang memiliki kompresi yang baik

(meskipun bersifat lossy) sehingga ukuran berkas bisa memungkinkan menjadi lebih kecil. MP3 memakai pengodean Pulse Code Modulation (PCM). MP3 mengurangi jumlah bit yang diperlukan dengan menggunakan model psychoacoustic untuk menghilangkan komponen-komponen suara yang tidak terdengar oleh manusia. MP3 memakai sebuah transformasi hybrid untuk mentransformasikan sinyal pada ranah waktu ke sinyal pada ranah frekuensi:

- Filter polyphase quadrature 32-band
- 36 atau 12 MDCT (modified discrete cosine transform), dengan ukuran dapat dipilih secara independen untuk sub-band 0...1 dan 2...31
- Postproses aliasing reduction

Standar MPEG-1 tidak menspesifikasikan secara spesifik cara melakukan encode MP3. Sebaliknya, algoritma decode serta format file didefinisikan secara spesifik. Yang ingin mengimplementasikan encoder MP3 harus membuat sendiri algoritma untuk menghilangkan bagian dari informasi pada file audio asal (atau pada representasi MDCT pada ranah frekuensi).

Karena itu, maka cara encode setiap encoder MP3 berlainan dan menghasilkan kualitas hasil yang berlainan juga. Hal yang harus diperhatikan adalah dari semua encoder yang ada, terdapat encoder yang bagus untuk bitrate tinggi maupun encoder yang bagus untuk bitrate rendah.

MP3 mempunyai beberapa batasan/limit:

- Bit rate terbatas, maksimum 320 kbit/s (beberapa encoder dapat menghasilkan bit rate yang lebih tinggi, tetapi sangat sedikit dukungan untuk mp3-mp3 tersebut yang memiliki bit rate tinggi)
- Resolusi waktu yang digunakan mp3 dapat menjadi terlalu rendah untuk sinyal-sinyal suara yang sangat transient, sehingga dapat menyebabkan noise.
- Resolusi frekuensi terbatas oleh ukuran window yang panjang kecil, mengurangi efisiensi coding
- Tidak ada scale factor band untuk frekuensi di atas 15,5 atau 15,8 kHz
- Mode jointstereo dilakukan pada basis per frame

Delay bagi encoder/decoder tidak didefinisikan, sehingga tidak ada dorongan untuk gapless playback (pemutaran audio tanpa gap). Tetapi, beberapa encoder seperti LAME dapat menambahkan metadata

yang baik tambahan yang memberikan informasi kepada MP3 player untuk mengatasi hal itu.

3. PEMBAHASAN

Metode MELFREQUENCYCEPSTRUM

Tujuan utama dari pemroses MFCC adalah menirukan perilaku dari pendengaran manusia. Adapun prosesnya sebagai berikut.

Frame Blocking

Dalam langkah ini sinyal wicara kontinu diblok menjadi frame-frame N sampel, dengan frame-frame berdekatan dengan spasi M ($M < N$). Frame pertama terdiri dari N sampel pertama. Frame kedua dengan M sampel setelah frame pertama, dan overlap dengan $N-M$ sampel. Dengan cara yang sama, frame ketiga dimulai $2M$ sampel setelah frame pertama (atau M sampel setelah frame kedua) dan overlap dengan $N-2M$ sampel. Proses ini berlanjut hingga semua wicara dihitung dalam satu atau banyak frame. Nilai tipikal untuk N dan M adalah $N = 256$ dan $M = 100$.

Windowing

Langkah berikutnya adalah pemrosesan dengan window pada masing-masing frame individual untuk meminimalisasi sinyal tak kontinu pada awal dan akhir masing-masing frame. Window dinyatakan sebagai $w(n)$, $1 \leq n \leq N$, dengan N adalah jumlah sampel dalam masing-masing frame, $x_1(n)$ adalah sinyal input dan hasil windowing adalah $y_1(n)$.

$$y_1(n) = x_1(n)w(n), \quad 0 \leq n \leq N - 1$$

Jenis window yang digunakan adalah window Hamming [4].

$$w(n) = 0.54 - 0.46 \cos \left[\frac{2\pi n}{N-1} \right], \quad 0 \leq n \leq N-1$$

dengan N adalah jumlah sampel.

Transformasi Fourier Cepat

Langkah pemrosesan berikutnya adalah transformasi fourier cepat/ fast fourier transform (FFT), FFT ini mengubah masing-masing frame N sampel dari domain waktu menjadi domain frekuensi. FFT adalah algoritma cepat untuk mengimplementasikan discretefourier transform (DFT) dengan didefinisikan pada kumpulan (set) N sampel, $\{x_k\}$, seperti berikut ini [7]. dengan, x_k = deretan aperiodik dengan nilai $N =$ jumlah sampel

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi i k n / N}, \quad n = 0, 1, 2, \dots, N-1$$

Mel-Frequency Wrapping

Studi psikofisikal menunjukkan bahwa persepsi manusia dari kandungan frekuensi suara pada sinyal wicara tidak mengikuti skala linier. Untuk masing-masing nada dengan frekuensi aktual, f dalam Hz, $pitch$ diukur dengan skala 'mel'. Skala mel -frequency adalah frekuensi linier berada dibawah 1000 Hz dan bentuk logaritmik berada diatas 1000 Hz. Sebagai titik referensi adalah $pitch$ dengan $tone$ 1 kHz, 40 dB diatas nilai batas ambang pendengaran, ini dinyatakan 1000 mel . Pendekatan persamaan untuk menghitung mel dalam frekuensi f (Hz) adalah [1][6].

$$mel(f) = 2595 \times \log_{10}(1 + f / 700)$$

Salah satu pendekatan simulasi spektrum yaitu menggunakan $filter$ bank, satu $filter$ untuk masing-masing komponen mel -frequency yang diinginkan. $Filter$ bank mempunyai respon frekuensi $bandpass$ segitiga dan jarak $bandwidth$ ditentukan oleh konstanta interval mel -frequency.

Cepstrum

Langkah selanjutnya yaitu mengubah spektrum log mel menjadi domain waktu. Hasil ini disebut mel frequency cepstrum coefficient (MFCC). Reprerentasi cepstral dari spectrum wicara memberikan reprerentasi baik dari sifat-sifat spektral lokal sinyal untuk analisis $frame$ yang diketahui. Karena koefisien mel spectrum adalah bilangan nyata. Dengan mengubahnya menjadi domain waktu menggunakan $discrete$ $cosinetransform$ (DCT). Jika koefisien spektrum daya mel hasilnya adalah c_1, c_2, \dots, c_K sehingga MFCC dapat dihitung, n adalah [8]

$$\tilde{c}_n = \sum_{k=1}^K (\log \tilde{S}_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \quad n = 1, 2, \dots, K$$

Vektor Kuantisasi

VQ atau vektor kuantisasi adalah proses memetakan

Pelatihan Vektor-Vektor

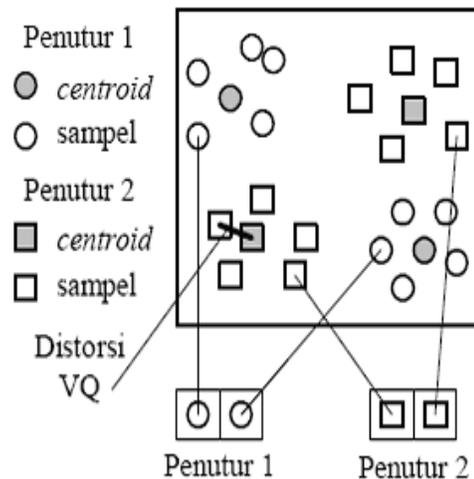
Selajutnya vektor-vektor akustik diekstraksi dari input wicara dari seorang penutur sebagai set pelatihan vektor-vektor. Sebagaimana penjelasan diatas, langkah penting berikutnya adalah membangun VQ $codebook$ dari penutur yang spesifik dengan

vektor-vektor dari ruang vektor besar menjadi jumlah terbatas daerah ruang vektor. Masing-masing daerah disebut kluster dan dapat direpresentasikan oleh pusatnya yang disebut $codeword$. Kumpulan dari semua $codeword$ - $codeword$ disebut $codebook$.

Dalam Gambar 5 menunjukkan konsep diagram untuk ilustrasi proses pengenalan. Hanya dua penutur dan dua dimensi dari ruang akustik ditunjukkan. Putaran-putaran mengacu pada vektor-vektor akustik dari penutur 1 dengan tanda lingkaran dan penutur 2 dengan tanda kotak. Dalam fasa pelatihan, VQ $codebook$ penutur-spesifik dibangkitkan untuk masing-masing penutur yang dikenali oleh pengklusteran vektor-vektor akustik pelatihan dari laki-laki atau wanita.

Hasil $codeword$ - $codeword$ ($centroid$) ditunjukkan dalam Gambar 1 dengan tanda lingkaran hitam dan tanda kotak hitam untuk penutur 1 dan 2. Jarak terdekat antara vektor $codeword$ dari $codebook$ disebut distorsi VQ. Dalam fasa pengenalan ini, input wicara dari suara tak dikenal adalah "vektor terkuantisasi" dengan menggunakan masing-masing $codebook$ yang dilatih dan jarak total distorsi VQ.

Penutur dengan VQ $codebook$ dan total distorsinya terkecil akan diidentifikasi.



Gambar 1 Formasi VQ Codebook antara penutur 1 dan penutur 2

menggunakan pelatihan vektor-vektor ini. Algoritma ini dikenal sebagai algoritma LBG (Linde, Buzo, dan Gray) [9], untuk kluster set L pelatihan vektor-vektor menjadi set M $codebook$ vektor-vektor. Algoritma ini Secara formal diimplementasikan dengan prosedur rekursif berikut ini:

1. Desain satu vektor $codebook$, ini adalah $centroid$

dari masukan set pelatihan vektor-vektor (karenanya tak diperlukan iterasi disini).

2. Gandakan ukuran *codebook* dengan membagi masing-masing *codebook* sekarang y_n sesuai dengan aturan dimana n berubah dari 1 ke ukuran *codebook* sekarang dan ϵ adalah parameter pembagi (*splitting*) (misalnya $\epsilon = 0.01$)

$$y_n^+ = y_n(1 + \epsilon)$$

$$y_n^- = y_n(1 - \epsilon)$$

3. Pencarian *neighbor*(tetangga) terdekat: untuk masing-masing pelatihan vektor, tentukan *codebook* dalam *codebook* yang terdekat dan menetapkan vektor-vektor tersebut yang berhubungan dengan sel (berhubungan dengan *codebook* terdekat).
4. *Centroid* terbarukan (*update*): mem-perbarui *codebook* dalam masing-masing sel dengan menggunakan *centroid* dari pelatihan vektor-vektor ini yang ditentukan untuk sel tersebut.
5. Iterasi 1: mengulangi langkah 3 dan 4 sampai jarak rata-rata jatuh dibawah nilai ambang.
6. Iterasi 2: mengulangi langkah 2, 3, dan 4 sampai *codebook* dengan ukuran M didesain. Algoritma LBG mendesain M vektor *codebook* dalam langkah ini. Langkah pertama dengan mendesain satu vektor *codebook*, kemudian menggunakan teknik pemecahan (*splitting*) pada *codebook* untuk menginialisasi pencarian untuk 2 vektor *codebook* dan melanjutkan proses pemecahan sampai M vektor *codebook* yang diinginkan akan dapat ditentukan.

Dalam Gambar 6 menunjukkan langkah detail algoritma LBG. Kluster vektor-vektor adalah *neighbor* terdekat yang menentukan masing-masing pelatihan vektor pada kluster yang berhubungan dengan

codebook terdekat. "Penentuan *centroid*" adalah prosedur *centroid* terkini. "Mengkomputasi D (distorsi)" yaitu menjumlahkan jarak semua pelatihan vektor-vektor pada *neighbor* terdekat dan menentukan apakah prosedurnya telah konvergen.

Algoritma K-Means

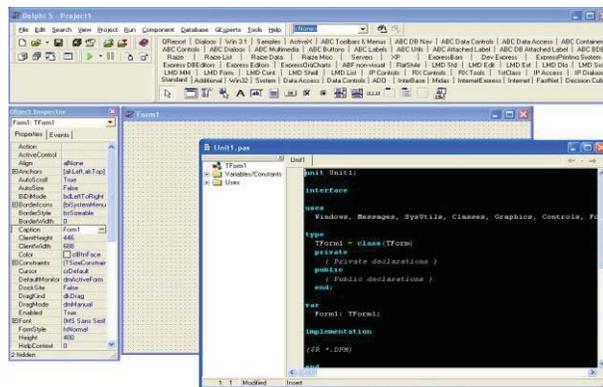
Algoritma *K-mean* adalah cara untuk mengkluster vektor-vektor pelatihan untuk mendapatkan vektorvektor ciri. Dalam algoritma ini vektor-vektor dikluster berdasarkan atribut menjadi K partisi. Ini menggunakan *K-mean* data dengan distribusi gaussian untuk mengkluster vektor-vektor tersebut. Tujuan *K-mean* adalah untuk meminimkan total varians intra-kluster, V , dimana ada K kluster k i $S_i, \dots, 3, 2, 1, =$ dan $i \mu$ adalah *centroid* atau titik *mean* dari semua titik.

Delphi

Delphi adalah sebuah [bahasa pemrograman](#) dan lingkungan

$$V = \sum_{i=1}^k \sum_{j \in S_i} |x_j - \mu_i|^2$$

pengembangan perangkat lunak. Produk ini dikembangkan oleh Borland (sebelumnya dikenal sebagai [Inprise](#)). Bahasa Delphi, yang sebelumnya dikenal sebagai object pascal (pascal dengan ekstensi [pemrograman berorientasi objek](#) (PBO/OOP)) pada mulanya ditujukan hanya untuk Microsoft Windows, namun saat ini telah mampu digunakan untuk mengembangkan aplikasi untuk Linux dan [Microsoft .NET framework](#) (lihat di bawah). Dengan menggunakan Free Pascal yang merupakan proyek opensource, bahasa ini dapat pula digunakan untuk membuat program yang berjalan di sistem operasi Mac OS X dan Windows CE.



Gambar1.2:Tampilan Delphi

Nada pada Musik

Nada dalam musik hanya ada 12 saja, dilambangkan dengan huruf A sampai G, yaitu C - C# - D - D# - E - F - F# - G - G# - A - A# - B - (Kembali ke) C. Antara nada ke nada berikutnya didefinisikan berjarak 1/2 (setengah), Sebagai contoh, nada C menuju C# berjarak setengah, C menuju D berjarak satu, C menuju D# berjarak satu setengah dan seterusnya. Nada dasar atau *triad* hanya terdiri 3 nada saja. Tiga nada dalam nada *triad* tersebut masing-masing dilambangkan sebagai I - II - III (maksudnya, seperti pada paduan suara I, suara II dan suara III). Pada prakteknya, lambang I, II dan III nanti akan diganti dengan lambang nada (huruf A sampai G tadi) apabila diperlukan untuk mendefinisikan nada yang diwakili oleh lambang I,II dan III. Ada 2 jenis nada *triad*, yaitu nada mayor dan nada minor.

1. Nada mayor terdiri dari 3 nada, nada I ke nada II berjarak dua, dan nada II ke nada III berjarak satu setengah.
2. Nada minor terdiri dari 3 nada, dimana nada I ke nada II berjarak satu setengah, dan nada II ke nada III berjarak dua.

Nada *triad* hanya terdiri 3 nada saja, sudah kesepakatan oleh musisi seluruh dunia bahwa nada yang berfungsi sebagai nada I digunakan sebagai nama nada, misalnya nada I adalah C maka nada yang terbentuk diberi nama C. Untuk membedakan penulisan nada mayor dan nada minor, nada Mayor memiliki lambang huruf **M** (huruf kapital), dalam penulisan nada, huruf M tidak dituliskan. Nada Minor memiliki lambang huruf **m** (huruf kecil), dalam penulisan nada lambang m ini dituliskan di belakang nama nada. Sehingga C, C#, D, D#, E, F, F#, G, G#, A, A#, dan B adalah nada Mayor. Sedangkan Cm, C#m, Dm, D#m, Em, Fm, F#m, Gm, G#m, Am, A#m, dan Bm adalah nada Minor.

Tinggi rendahnya nada dan nada

Salah satu karakteristik yang paling penting dari sebuah nada adalah *pitch*. *Pitch* didefinisikan sebagai tinggi atau rendah sebuah nada yang dinyatakan dalam besaran frekuensi. Nilai dari *pitch* ini sangat tergantung pada frekuensi dasar. Sebagai contoh, frekuensi dasar untuk A4 adalah 440 Hz. Pada sebuah oktaf, interval antara dua buah nada (not) atau nada adalah digandakan dari frekuensi dasar. Sebagai contoh : (1)

$$f(A5) = 2f(A4)$$

Pada *the western well-tempered scale*, frekuensi nada n1 diketahui dari frekuensi nada n2, diukur dalam semi-tones seperti ditunjukkan Persamaan (2).

$$f(n_1) = f(n_2).2^{\frac{1}{12}}$$

Simbol S menunjukkan perbedaan antara n2 dan n1 di dalam *semi-tones*. Ketika sebuah not atau nada dimainkan dengan alat musik, tidak hanya frekuensi dasar yang dihasilkan, tetapi juga terdapat sekumpulan frekuensi lain yang disebut dengan frekuensi harmonik. Frekuensi harmonik umumnya mempunyai perbedaan amplitudo dari frekuensi dasar. Bentuk gelombang amplitudo dapat dilihat dalam karakteristik spektrum yang disebut dengan *timbre* pada sebuah alat musik. Persamaan dari frekuensi harmonik ditunjukkan Persamaan (3).

$$f(h_i) = (i + 1)f(h_0)$$

$$f(h_i) = (i + 1)f(h_0)$$

Dimana f(hi) adalah frekuensi harmoni urutan ke i dari frekuensi dasar f(ho).

Nada memiliki empat parameter penting yaitu:

1. *Pitch* : menunjukkan tinggi atau rendah suatu nada (analogi frekuensi).
2. *Durasi/ritme* : menunjukkan panjang atau pendek suatu nada dimainkan (analogi lama waktu).
3. *Loudness* : menunjukkan keras atau lemah suatu nada (analogi amplitudo).
4. *Timbre* : warna nada atau kualitas suatu nada. Sebagai contoh, nada yang dihasilkan oleh alat musik seperti gitar, piano, trumpet, violin, dan lain-lain.

Pengembangan Sistem Cerdas (*Expert System*)

Analisis Kebutuhan

Dalam pembuatan suatu program aplikasi dibutuhkan adanya suatu analisis yang tepat tentang hal-hal yang dibutuhkan, setelah sebelumnya dilakukan proses perencanaan sehingga didapat suatu program aplikasi yang sesuai dengan kebutuhan. Setelah proses analisis dilakukan dengan benar, maka dapat diketahui kebutuhan sistem dengan tepat.

Identifikasi Masalah

Dari latar belakang yang telah disebutkan sebelumnya, diketahui bahwa inti permasalahan yang muncul adalah Dalam permainan gitar, kunci atau *chord* gitar sangat penting dipelajari dan diingat sehingga seorang pemain gitar dapat memainkan gitar dengan baik. Proses belajar gitar secara otodidak atau belajar sendiri (tampa les atau sekolah khusus) sangatlah sulit, adapun beberapa masalah yang dihadapi adalah:

1. Seorang pemain gitar pemula kesulitan memprediksi *chord* gitar dari lagu MP3.

2. Proses ekstraksi *chord* gitar dari file MP3 memerlukan waktu yang lama

Kebutuhan Software

Dalam penelitian ini, pendekatan yang dilakukan adalah dengan penggunaan aplikasi sebagai media untuk dapat mengekstrak chord lagu dari MP3. Untuk membuat aplikasi tersebut menarik, maka konsep yang digunakan adalah dengan system ekstrak. Untuk

Perancangan Sistem

Dalam merancang sistem yang dalam hal ini berupa

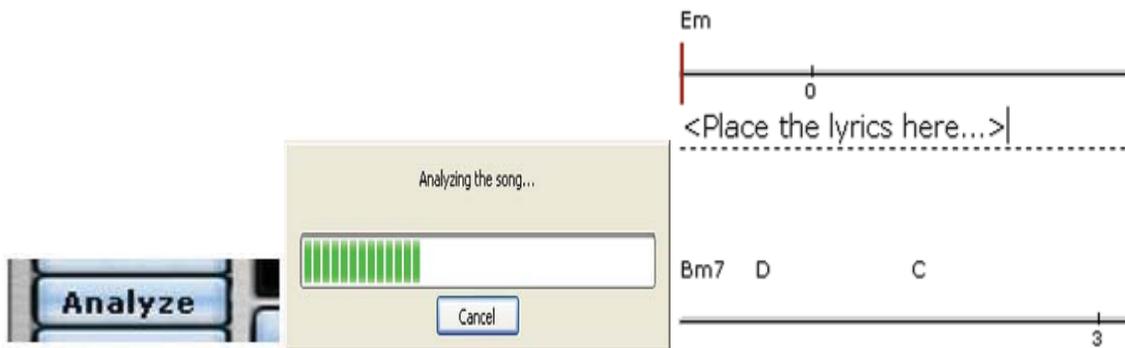
itu, digunakanlah Borland Delphi sebagai *software* pembangun aplikasi ekstraksi *chord* lagu dari mp3.

System requirements

- a. PC or 100% compatible
- b. Pentium II processor 600 MHz or higher
- c. Windows 98SE/ME/NT/2000/XP
- d. Minimum 128 Mb RAM
- e. Soundcard/CD-ROM/Monitor/Interne *aplikasi*, harus berpedoman pada karakteristik dan unsur:



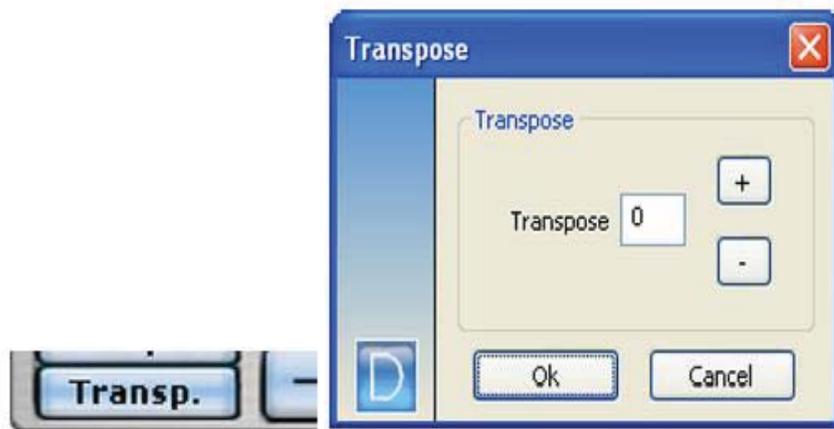
Gambar 3. Mengambil (load) lagu MP3 dengan format file type MP3, CDA, WAV, WMA, RAW, AVI, VOX, Ogg Vorbis, MPC, AIFF.



Gambar 4. Analisa dan Mengekstrak lagu Mp3 menjadi *chord* gitar



Gambar 5. Main(Play) Music MP3



Gambar 6. Nada Lagu

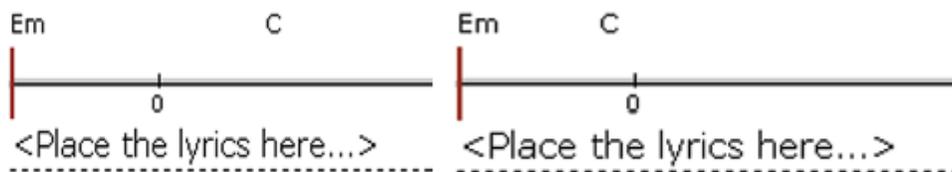


Gambar 7. Tampilan Chord Gitar

Memindahkan chord

Ini tidak mungkin untuk memindahkan senar berbeda antara baris. Jika anda ingin untuk melakukannya,

terlebih dahulu harus menghapus chord dan kemudian memasukkan chord pada baris yang dikehendaki. Terlihat pada gambar dibawah ini.

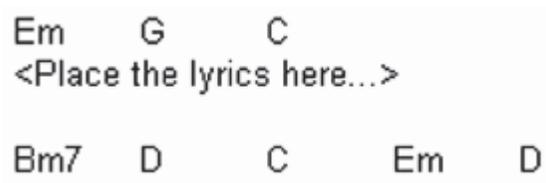


Gambar 8. Merubah dan menganalisa lagu

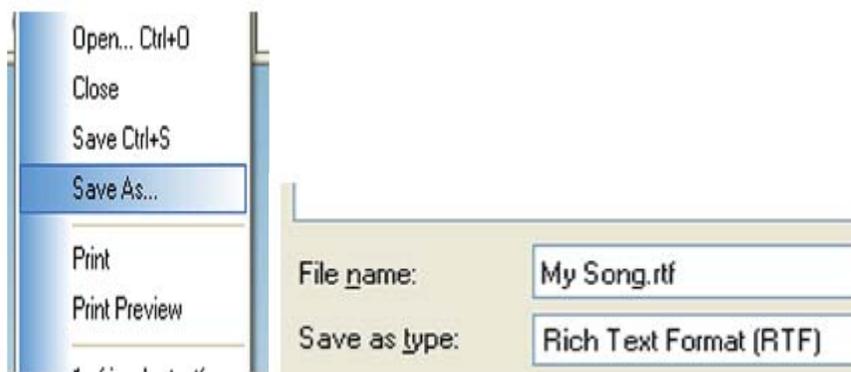
Menganalisis hasil ekspor ke MS Word

lagu harus diambil dan dianalisis sebelum diekspor ke

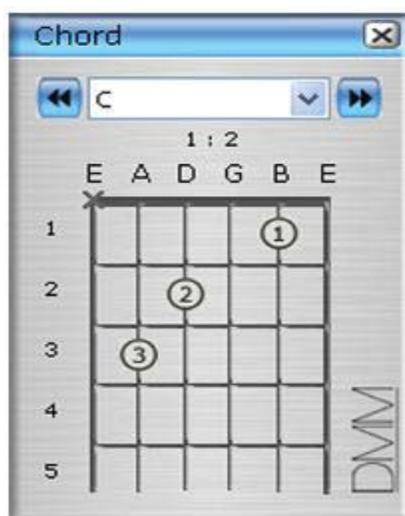
MS Word kompatibel format *Rich Text Format* (RTF).



Gambar 9. Edit Chord Dari Menu Swict pilih Edit view



Gambar 10. Menyimpan kedalam Folder



Gambar 11. Kunci Dialog Chord

Implementasi dan Pengukuran Software

Setelah aplikasi dibuat, maka langkah selanjutnya adalah penerapan sistem tersebut terhadap objek penelitian, yaitu dalam hal ini kelompok pemusik mahasiswa yang ingin belajar mengetahui chord gitar dari MP3.

Metode Implementasi

Sesuai dengan batasan objek penelitian, maka penulis menerapkan aplikasi ekstrak yang dibuat ini terhadap pemain gitar pemula kesulitan memprediksi chord gitar dari lagu MP3 Untuk itu, penulis membagi tahapan implementasi ini menjadi beberapa bagian sebagai berikut:

1. Menetapkan lag MP3 dengan format yang sudah di sebutkan idatas
2. Observasi lapangan untuk membuktikan apakah selama ini pemusik pemula kesulitan memprediksi kunci-kunci gitar dari lagu MP3,

3. Penerapan aplikasi sebagai alat bantu mengekstrak lagu dari MP3,
4. Melakukan survei untuk mendapatkan data setelah implementasi (*post-test*) melalui pengisian kuesioner oleh pemain musik yang mengamati perilaku anak didik.
5. Melakukan analisa hasil pengukuran penelitian

4. KESIMPULAN

Dari hasil penelitian ini yang dilakukan mulai dari tahap awal hingga pengujian penerapan aplikasi ekstraksi *chord* gitar dari MP3 dengan menggunakan Metode *Mell-Frequence Cepstral Coefficiteins* ini, didapatkan kesimpulan bahwa proses ekstrasi *chord* akan mencapai hasil yang optimal jika didukung dengan pendekatan dan metode yang sesuai, menarik dan interaktif. Selama ini, pemusik pemula belum mengetahui chord gitar dari lagu MP3. Berdasarkan hasil penelitian, penerapan aplikasi system cerdas ini

dapat membawa efek positif dalam proses pengetahuan akan *chord* gitar, namun terdapat beberapa hal yang perlu perluas bagi pengembangan sistem ini antara lain:

1. Dari variasi aplikasi yang ada, dapat dikembangkan lagi menjadi semua *chord* yang tidak berlaku untuk gitar dengan pengembangan tingkat kesulitan dan pengembangan *environment* yang lebih luas.
2. Variasi Type file dr MP3 bisa ditambah untuk menambah pengenalan system ekstrak dari software tersebut.
3. Selain dapat mengekstraksi software tersebut dapat menyalin kedalam file teks.
4. Apabila *software* ini akan diterapkan untuk pembelajaran yang lain, hendaknya pembelajaran dapat diterapkan ke dalam *template* yang dapat diubah isinya sehingga pokok bahasan dan segmen pengguna dapat diperluas

Daftar Pustaka

- Ali Mustofa, Ali. 2007. Sistem Pengenalan Penutur dengan Metode *Mel-frequency Wrapping*. Jurnal Teknik Elektro Vol. 7, No. 2, September 2007: 88 – 96.
- Indriyawan, Eko. 2007. Sejarah Delphi, Delphi 2007 for Win32.
- Mp3, <http://id.wikipedia.org/wiki/Mp3>
- Muttaqin, Muhammad. 2008. Pengertian Musik, Buku Sekolah Elektronik. Direktorat Jenderal Manajemen Pendidikan Dasar dan Menengah Departemen Pendidikan Nasional, Seni Musik Klasik.