

PEMISAHAN CHORUS PADA MUSIC MP3 MENGGUNAKAN KOEFISIEN KORELASI 2-D BERBASIS DISCRETE COSINE TRANSFORM (DCT) DAN K-NEAREST NEIGHBOR (K-NN)

SEPARATION OF CHORUS ON MP3 MUSIC USING 2-D CORRELATION COEFFICIENT BASED ON DISCRETE COSINE TRANSFORM (DCT) AND K-NEAREST NEIGHBOR (K-NN)

Muhammad Wahyu Setiawan¹, Ledy Novamizanti, S.Si., M.T.², I N Apraz Ramatryana, S.T., M.T.³

Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹mwahyusetiawan30@gmail.com, ²ledyaldn@telkomuniversity.ac.id, ³ramatryana@gmail.com

Abstrak

Penelitian tentang metode pencarian judul lagu dengan input suara senandung manusia atau *humming* membutuhkan *chorus* dari sebuah lagu untuk mendapatkan pola nada yang unik dari sebuah lagu. Sebelumnya, proses pemisahan *chorus* masih dilakukan secara manual sehingga membutuhkan waktu yang lama. Sehingga, diperlukan sebuah metode untuk menentukan posisi *chorus* dan memisahkannya secara otomatis dengan memanfaatkan pengolahan sinyal audio.

Pada penelitian ini, penulis merancang metode penentuan posisi *chorus* dengan menggunakan nilai Koefisien Korelasi 2-Dimensi (KK2-D) dan menambahkan proses klasifikasi K-Nearest Neighbor (K-NN). Proses pertama adalah input file audio, selanjutnya *preprocessing*, *framing*, *windowing*, transformasi *Discrete Cosine Transform (DCT)*, perhitungan KK2-D, dan klasifikasi K-NN. Pada penelitian ini, data yang digunakan sebanyak 25 data lagu yang terdiri dari 5 *genre*. Pengujian yang dilakukan adalah menganalisis pengaruh ukuran *frame*, pengaruh jenis *window*, nilai K dari K-NN, dan jenis jarak dari K-NN. Hasil akurasi rata-rata dari 25 data lagu adalah 95% dengan parameter terbaik terdiri dari ukuran *frame* 1 detik, jenis *window rectangular*, nilai K sebesar 5, dan jarak cosine. Waktu rata-rata proses untuk satu lagu adalah 0,112 detik.

Kata kunci : Sinyal audio, chorus, DCT, koefisien korelasi 2-D.

Abstract

Research about song title finder method with humming sound input need chorus part of the song to get unique pattern from a song. Chorus separation process still done manually so it will take a long time. This problem become the background, we need a method to finding the chorus position and separate it automatically by using audio signal processing.

In this research, author designing a chorus position finder method using 2-D Coefficient Correlation (KK2-D) and adding classification process K-Nearest Neighbor (K-NN). First process is inputting the audio file, next are *preprocessing*, *framing*, *windowing*, *Discrete Cosine Transform (DCT)* transform, KK2-D calculation, and K-NN classification. In this research, author used 25 songs from 5 different genres. Test that have been done are analyzing frame size affection, window type affection, K-value from K-NN. Accuracy result from 25 songs is 95% with the best parameters are with 1 second frame, rectangular window type, K value of 5, and cosine distance. Average computation time for one song is 0.112 second.

Keywords: Audio signal, chorus, DCT, 2-D correlation coefficient.

1. Pendahuluan

Di masa sekarang perkembangan teknologi telekomunikasi semakin pesat dan menyeluruh. Teknologi telekomunikasi tidak hanya serta merta mengirimkan informasi dari satu titik ke titik yang lain, tetapi meluas ke dunia entertainment contohnya dunia musik. Pengolahan sinyal informasi ini sangat mencakup luas, salah satunya yaitu mengidentifikasi sinyal informasi pada lagu. Lagu dijadikan sebagai objek yang utama, hal ini disebabkan perkembangan music yang begitu pesat juga.

Penelitian sebelumnya tentang klasifikasi genre musik menggunakan Hidden Markov Model [1] dan penelitian pengenalan suara [2]. Kemudian metode pencarian judul lagu dengan input suara senandung manusia atau humming yang telah diteliti menggunakan Discrete Cosine Transform (DCT) [3], Fast Fourier Transform (FFT) [4], Linear Predictive Coding (LPC) [5], dan Mel-Frequency Cepstral Coefficient (MFCC) [6]. Pada penelitian tersebut dibutuhkan database yang berupa verse dan reff dari lagu. Pada penelitian tersebut menggunakan 100 data lagu yang dipotong menjadi potongan verse dan reff secara manual menggunakan software Cool Edit. Hal ini menjadi masalah apabila jumlah database verse dan reff yang dibutuhkan semakin banyak

sehingga diperlukan penelitian lebih lanjut untuk melakukan proses pemotongan atau pemisahan verse dan reff secara otomatis dengan menganalisis sinyal dari file music pada MP3 lagu tersebut.

Pada penelitian [7], letak verse dan reff ditentukan dengan menggunakan metode FFT. Tetapi letak awal verse dan reff dipisahkan secara manual pada tampilan GUI. Metode pemisahan letak verse dan chorus yang sudah dikerjakan adalah menggunakan metode FFT, LPC [8], dan discrete wavelete transform (DWT) [9]. Dari semua metode tersebut merupakan analisis domain frekuensi dari sinyal audio dan menghasilkan hasil yang buruk apabila menggunakan file audio dengan genre rock. Untuk mengatasi masalah tersebut, penulis menggunakan perhitungan korelasi silang antar kumpulan frame yang dijadikan referensi sebagai chorus terhadap kumpulan frame seluruh lagu. Proses selanjutnya adalah melakukan proses penentuan posisi chorus dari hasil korelasi silang tersebut dengan penambahan proses klasifikasi K-NN sehingga didapatkan posisi detik pada lagu yang diindikasikan sebagai chorus.

2. Dasar Teori

2.1 Discrete Cosinus Transform (DCT)

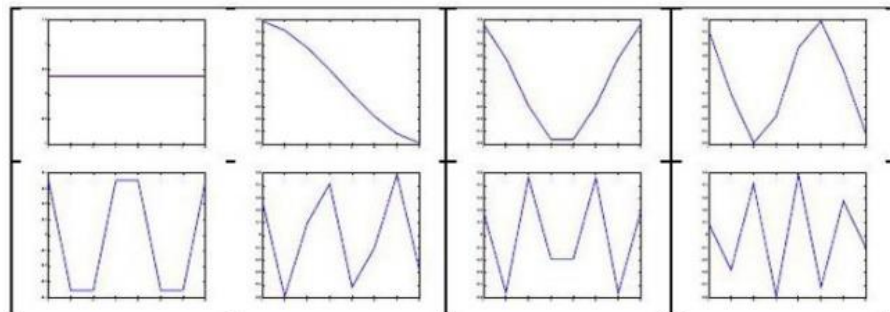
Discrete Cosine Transform (DCT) dapat digunakan untuk mengubah sebuah sinyal menjadi komponen frekuensi dasarnya. *Discrete Cosine Transform* dari sederet n bilangan real $s(x)$, $x = 0, \dots, n-1$, dirumuskan sebagai berikut [8]:

$$S(u) = C(u) \sqrt{\frac{2}{n}} \sum_{x=0}^{n-1} s(x) \cos\left(\frac{(2x+1)u\pi}{2n}\right), \quad (1)$$

dimana $u = 0, \dots, n-1$, dimana

$$C(u) = \begin{cases} 2^{-\frac{1}{2}}, & \text{untuk } u = 0 \\ 1, & \text{untuk } u \text{ lainnya.} \end{cases} \quad (2)$$

Setiap elemen dari hasil transformasi $S(u)$ merupakan hasil *dot product* atau *inner product* dari masukan $s(x)$ dan basis vektor. Faktor konstanta dipilih sedemikian rupa sehingga basis vektornya orthogonal dan ternormalisasi. DCT juga dapat diperoleh dari produk vektor (masukan) dan $n \times n$ matriks orthogonal yang setiap barisnya merupakan basis vektor. Delapan basis vektor untuk $n = 8$ dapat dilihat pada Gambar 1. Setiap basis vektor berkorespondensi dengan kurva sinusoid frekuensi tertentu.



Gambar 1 Delapan Basis Vektor Untuk DCT dengan $n = 8$

Barisan $s(x)$ dapat diperoleh lagi dari hasil transformasinya $S(u)$ dengan menggunakan *Invers Discrete Cosine Transform* (IDCT), yang dirumuskan sebagai berikut :

$$s(x) = \sqrt{\frac{2}{n}} \sum_{u=0}^{n-1} S(u) C(u) \cos\left(\frac{(2x+1)u\pi}{2n}\right), \quad (3)$$

dengan $u = 0, \dots, n-1$, dimana

$$C(u) = \begin{cases} 2^{-\frac{1}{2}}, & \text{untuk } u = 0 \\ 1, & \text{untuk } u \text{ lainnya.} \end{cases} \quad (4)$$

Persamaan diatas menyatakan s sebagai kombinasi linier dari basis vektor. Koefisien adalah elemen transformasi S , yang mencerminkan banyaknya setiap frekuensi yang ada didalam masukan s .

Discrete Cosine Transform merepresentasikan sebuah citra dari penjumlahan sinusoida dari magnitude dan frekuensi yang berubah-ubah. Sifat dari DCT adalah mengubah informasi citra yang signifikan dikonsentrasikan hanya pada beberapa koefisien DCT. Oleh karena itu DCT sering digunakan untuk kompresi citra seperti pada JPEG [3].

Untuk sinyal x dari panjang N , dan dengan δ_{kl} delta Kronecker, transformasi didefinisikan oleh:

$$y(k) = \sqrt{\frac{2}{N}} \sum_{n=1}^N x(n) \cos\left(\frac{\pi}{4N} (2n-1)(2k-1)\right), \quad (5)$$

Deret diberi indeks dari $n = 1$ dan $k = 1$. Semua varian dari DCT adalah kesatuan (atau, ekuivalen, ortogonal): Untuk menemukan inversnya, tukar k dengan n pada setiap definisi [4].

2.2 K-Nearest Neighbor (K-NN)

K-NN merupakan salah satu metode klasifikasi pada citra yang berdasarkan ciri-ciri data pembelajaran (data latih) yang paling mendekati objek. Ciri direpresentasikan dengan ukuran jarak yang diolah dalam hitungan matematis. Pada K-NN dihitung nilai jarak antara titik yang merepresentasikan data pengujian dengan semua titik yang merepresentasikan data latihnya. Untuk menghitung jarak antar tetangga digunakan beberapa cara, diantaranya [9]:

1. City Block atau Manhattan Distance

City block distance juga disebut sebagai *manhattan distance* atau *absolute distance*. *City block distance* menghitung nilai perbedaan absolut dari dua vektor.

$$L_1(X, Y) = \sum_{i=1}^d |X_i - Y_i|, \quad (6)$$

dimana X_i merupakan data latih, Y_i merupakan data uji, i adalah variable data, dan d adalah besar dimensi data.

2. Euclidean Distance

Euclidean Distance menghitung jarak data latih dan data tes, Berikut adalah rumus untuk menghitung Euclidean Distance:

$$L_1(X, Y) = \sqrt{\sum_{i=1}^d (X_i - Y_i)^2}, \quad (7)$$

3. Cosine Distance

Pada Correlation Distance, titik-titik dianggap sebagai barisan nilai. Berikut adalah rumus untuk menghitung Cosine Distance:

$$\cos(d_i, d_j) = \frac{\sum_k a_{i,k} \cdot a_{j,k}}{\sqrt{\sum_k a_{i,k}^2} \cdot \sqrt{\sum_k a_{j,k}^2}}, \quad (8)$$

4. Correlation Distance

Pada Correlation Distance, titik-titik dianggap sebagai barisan nilai. Berikut adalah rumus untuk menghitung Correlation Distance:

$$\frac{\sum_{k=1}^n (X_{ik} - \bar{x}_i)(X_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (X_{ik} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^n (X_{jk} - \bar{x}_j)^2}}, \quad (9)$$

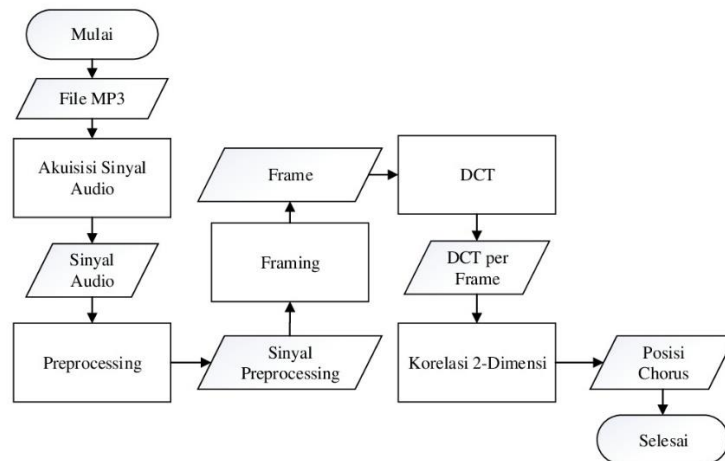
dimana $\bar{x}_i = \frac{1}{n} \sum_{k=1}^n x_{ik}$ dan $\bar{x}_j = \frac{1}{n} \sum_{k=1}^n x_{jk}$.

Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Kasus khusus dimana klasifikasi diprediksikan berdasarkan data latih yang paling dekat (dengankatalain, $k = 1$) disebut algoritma *nearest neighbor*. KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap data latih yang memiliki banyak *noise* dan efektif apabila data latihnya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat), data latih berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap pengujian pada keseluruhan data latihnya.

3. Perancangan Sistem

3.1 Diagram Alir Perancangan Sistem

Metode penentuan posisi chorus yang dirancang menggunakan metode *Discrete Cosine Transform* (DCT) untuk ekstraksi ciri, korelasi dua dimensi untuk mendapatkan Koefisien Korelasi 2-Dimensi (KK2-D), dan klasifikasi *K-Nearest Neighbor* (K-NN). Proses dimulai dengan memilih file MP3 dan selanjutnya didapatkan data sinyal audio. Proses selanjutnya adalah *preprocessing* melakukan proses pemotongan sinyal audio menjadi potongan kecil yang disebut *frame* dan proses ini disebut proses *Framing*. Setiap *frame* dilakukan proses transformasi DCT untuk mendapatkan nilai koefisien DCT. Koefisien DCT disini merupakan ciri yang membedakan satu *frame* dengan *frame* yang lain. Proses terakhir adalah proses penentuan posisi *chorus* dengan menggunakan proses korelasi 2-D dan klasifikasi K-NN. Diagram alir perancangan sistem dapat dilihat pada Gambar 2.



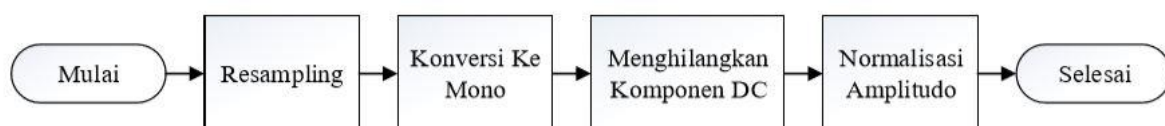
Gambar 2 Diagram Alir Perancangan Sistem

3.2 Sinyal Audio

Sinyal audio yang digunakan adalah sinyal diskrit hasil dari proses pembacaan pada bahasa pemrograman matriks dengan frekuensi sampling $f_s = 44100$ sampel/detik.

3.2.1 Pre-processing

Proses *pre-processing* terdiri dari proses resampling, konversi sinyal ke sinyal satu kanal atau sinyal mono, menghilangkan komponen DC, dan normalisasi amplitudo. Alur proses dijelaskan pada Gambar 3.



Gambar 3 Flowchart Pre-Processing

3.2.2 Resampling

Proses resampling adalah proses mengubah jumlah *sampel* data dalam satu detik atau menurunkan nilai frekuensi sampling yang awalnya sesuai f_s dari file MP3 menjadi f_s sebesar 8000 sampel/detik.

3.2.3 Stereo to Mono

Proses *stereo to mono* adalah proses konversi kanal dari audio MP3 menjadi satu kanal apabila kanal dari audio MP3 termasuk kanal *stereo*. Proses konversi dilakukan dengan menghitung rata-rata dari kedua kanal.

3.2.4 DC Removal

Proses ini adalah menghilangkan komponen DC dari sebuah sinyal audio. Proses dilakukan dengan cara mengurangkan nilai sinyal dengan nilai rata-rata dari sinyal tersebut.

3.2.5 Normalisasi Amplitudo

Proses normalisasi amplitudo bertujuan untuk menyamakan besar amplitudo dari sinyal agar ada direntang -1 dan 1 sehingga semua sinyal audio berada di level daya yang sama. Proses ini dilakukan dengan cara melakukan pembagian semua data sampel terhadap nilai maksimal dari semua data sampel.

3.2.6 Framing

Proses *framing* adalah proses pemotongan sinyal audio menjadi potongan kecil dengan ukuran yang sama yaitu sebanyak N_{Window} . Tujuan dari proses *framing* agar proses sinyal menjadi lebih fokus pada jumlah data yang terbatas.

3.2.7 Windowing

Windowing didefinisikan sebagai fungsi yang berguna untuk mengalikan sinyal terpotong yang *discontinue* terhadap fungsi *window*, hal ini bertujuan untuk mengembalikan sinyal *discontinue* menjadi sinyal yang *continue*. Proses *windowing* ini bertujuan untuk mengurangi terjadinya *aliasing* yang merupakan suatu efek dari timbulnya sinyal baru yang memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek tersebut dapat terjadi karena rendahnya jumlah sampling rate atau karena proses *frame blocking* yang menyebabkan sinyal menjadi *discontinue*. Ada beberapa teknik window yang biasa digunakan, teknik tersebut diantaranya adalah *Rectangular window*, *Hamming window* dan *Hanning window*.

Tabel 1 Persamaan *windowing*

Jenis <i>Window</i>	Persamaan
<i>Rectangular</i>	$w(n) = 1$
<i>Haming</i>	$w(n) = 0,54 - \left(0,46 \times \cos\left(\frac{2\pi n}{N-1}\right)\right)$,
<i>Hanning</i>	$w(n) = 0,5 - \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$,

3.2.8 Discrete Cosine Transform

Proses transformasi DCT bertujuan untuk mendapatkan ciri frekuensi dari tiap frame, sehingga panjangnya data sampel untuk proses DCT adalah sesuai dengan ukuran frame yaitu $N_{DCT} = N_{Window}$.

3.2.9 Algoritma Penentuan Posisi

Algoritma penentuan posisi menggunakan *High Frequency Content* (HFC) dengan perhitungan sesuai persamaan 8.

$$HFC(k, frame) = k \times x(k, frame), \quad (10)$$

4. Hasil Percobaan dan Analisa

4.1 Skenario 1 Pengujian dan Analisis Pengaruh Ukuran *Frame*

Pada skenario ini, penulis menggunakan *frame* 250ms, 500ms dan 1000ms. Adapun hasil yang diperoleh yaitu :

Tabel 2 Hasil Koefisien Korelasi 2 Dimensi DCT-KNN

Ukuran <i>frame</i> (detik)	Nilai KK-2D (detik)	
	<i>Chorus 2</i>	<i>Chorus 3</i>
0,25	0,61	0,54
0,5	0,67	0,60
1	0,83	0,75

Tabel 3 Ranking Posisi Pengujian Ukuran *Frame*

Ukuran <i>frame</i> (detik)	Ranking Posisi	
	<i>Chorus 2</i>	<i>Chorus 3</i>
0,25	405	289
0,5	170	187
1	13	18

Tabel 4 Waktu Proses Pengujian Ukuran *Frame*

Ukuran Frame (Detik)	Waktu (detik)
0,25	3,88
0,5	2,27
1	1,44

Dari pengujian skenario 1 dengan jumlah frame 1 detik didapat hasil pada nilai koefisien korelasi 2D terlihat pada Tabel 2 adalah semakin besar nilai korelasi maka hasil pengujian pada posisi *chorus* semakin baik dan hasil pada *ranking* posisi yang paling baik pada pengujian *frame* 1 detik pada Tabel 3 adalah semakin kecil nilai *ranking* maka semakin baik nilainya. Kemudian dari Tabel 4 didapatkan hasil rata rata waktu proses penentuan *chorus* paling baik pada frame 1 detik.

4.2 Skenario 2 Pengujian dan Analisis Pengaruh Jenis Window

Pada skenario kedua akan dilakukan *windowing*, dengan 3 jenis *windowing* yaitu *rectangular*, *hamming*, *hamming*, menggunakan ukuran *frame* 1000ms. Adapun hasil yang didapat adalah:

Tabel 5 Hasil Koefisien Korelasi 2 Dimensi DCT-KNN

Jenis Window	Nilai KK-2D (detik)	
	<i>Chorus 2</i>	<i>Chorus 3</i>
<i>Rectangular</i>	0,63	0,55
<i>Hamming</i>	0,60	0,51
<i>Hanning</i>	0,60	0,51

Dari hasil uji yang dilakukan, dapat dilihat jenis window mendapatkan rata-rata hasil yang mirip, jadi dapat disimpulkan semua jenis window bagus terhadap korelasi 2 dimensi

Tabel 6 *Ranking* Posisi Pengujian Ukuran *Frame*

Ukuran <i>frame</i> (detik)	<i>Ranking</i> Posisi	
	<i>Chorus 2</i>	<i>Chorus 3</i>
<i>Rectangular</i>	13,08	18,68
<i>Hamming</i>	20	24,76
<i>Hanning</i>	19,4	24,24

Dari hasil uji yang dilakukan, didapatkan hasil rata-rata *ranking* terendah yang merupakan hasil paling baik terdapat pada jenis window *rectangular*.

Tabel 7 Waktu Proses Pengujian Ukuran *Frame*

Ukuran Frame (Detik)	Waktu (detik)
<i>Rectangular</i>	3,64
<i>Hamming</i>	4,69
<i>Hanning</i>	3,84

Dari hasil uji yang dilakukan, tidak terdapat perbedaan yang signifikan antar jenis window jadi dapat disimpulkan semua jenis window terhadap waktu proses berbanding lurus atau sama.

4.3 Skenario 3 Pengujian dan Analisis Sistem Pemisahan Reff Tanpa Referensi (otomatis)

Dalam skenario ini dilakukan pengujian pengaruh ukuran frame pada proses ekstraksi ciri. Dalam pengujian digunakan data 25 lagu yang dikelompokkan menjadi 5 genre yaitu pop, EDM, funk, rock, dan hip-hop. Dari hasil pengujian, dilakukan analisis akurasi. Pada pengujian ini beberapa nilai yang diubah adalah

1. Ukuran *frame*, digunakan 250ms, 500ms, dan 1000ms.
2. Jenis *Window*, digunakan *rectangular*, *hamming*, dan *hanning*.

3. Nilai k pada K-NN, digunakan $k=1$, $k=3$, dan $k=5$;
4. Jarak Distance pada K-NN, digunakan *euclidean*, *correlation*, dan *cosine*

Pada Skenario otomatis ini didapatkan bahwa ukuran frame terbaik pada saat frame berukuran 1 detik, jenis window terbaik saat menggunakan window rectangular, nilai K-NN yang terbaik pada K yang digunakan adalah 5, dan jarak yang terbaik adalah *cosine*.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pada penelitian ini, telah dirancang suatu sistem untuk menentukan posisi *chorus* dari sinyal lagu sebuah file musik MP3, DCT sebagai transformasi dan ekstraksi ciri, dan klasifikasi K-NN. Berdasarkan hasil pengujian pada sistem diperoleh kesimpulan sebagai berikut:

1. Semakin besar ukuran *frame*, maka semakin banyak sampel data pada satu *frame* sehingga banyak data sampel DCT juga semakin banyak dan menghasilkan nilai KK2-D yang lebih tinggi. Waktu komputasi berbanding terbalik dengan ukuran *frame* karena semakin kecil ukuran *frame* menyebabkan semakin banyak nilai yang dihitung nilai KK2-D nya. Pada saat dilakukan pengujian sistem skenario 1 dengan penggunaan *frame* 250ms, 500ms dan 1000ms di dapat hasil sinyal audio yang optimal pada saat *frame* 1000ms.
2. Jenis *window rectangular* lebih baik dibandingkan *hamming* dan *hanning*.
3. Akurasi sistem pada penelitian ini adalah 95% dengan rata-rata waktu proses sebesar 0,11 detik dengan parameter terbaik terdiri dari ukuran *frame* 1 detik, jenis *window rectangular*, nilai K pada K-NN sebesar 5, dan jarak *cosine*.

5.2 Saran

Berikut ini saran untuk pengembangan penelitian :

1. Pada penelitian ini fokus untuk menguji metode K-NN sehingga digunakan data pada penelitian sebelumnya yang sebanyak 25 data. Untuk menguji performansi sistem sebaiknya dilakukan penambahan jumlah data pada *database*.
2. Komponen sebuah lagu terdiri dari *chord* yang terdapat pada instrumen dan lirik pada *vocal* dari lagu. Dengan menambahkan proses pemisahan *vocal* dan *unvocal* dapat lebih fokus untuk menentukan pola nada dari lagu tersebut pada *chorus*.

5

Daftar Pustaka

- [1] I. Ikhsan, L. Novamizanti, I.N.A. Ramatryana, "Automatic musical genre classification of audio using Hidden Markov Model", 2nd International Conference on Information and Communication Technology (ICoICT), 2014.
- [2] R.R. Juniansyah, R. Magdalena, L Novamizanti, Perancangan Sistem Pengenalan Suara Dengan Metode Linear Predictive Coding, eProceedings of Engineering, 4 (1), 2017
- [3] A. Archamadi, "Analisis dan Simulasi Identifikasi Judul Lagu dari Senandung Manusia Menggunakan Ekstraksi Ciri Discrete Cosine Transform". Bandung : Universitas Telkom.
- [4] I. Y. Kurniawan, "Analisis dan Simulasi Identifikasi Judul Lagu Melalui Senandung Manusia Menggunakan Ekstraksi Ciri Linier Predictive Coding". Bandung : Universitas Telkom.
- [5] O. G. Filemon, "Perancangan dan Simulasi Pemisahan Reff Lagu dengan Metode Fast Fourier Transform". 2017.
- [6] S. A. Armando, "Analisis dan Simulasi Pencarian Verse dan Reff Lagu Pada Musik Digital Dengan Metode Linear Predictive Coding". 2017.
- [7] F. Patriandhika, "Simulasi dan Analisis Pencari Reff dan Verse Lagu Pada Musik Digital Dengan Metode Korelasi". Bandung : Universitas Telkom
- [8] I. Iwut, G. Budiman, L. Novamizanti, Optimization Of Discrete Cosine Transform-Based Image Watermarking by Genetics Algorithm, Indonesian Journal of Electrical Engineering and Computer Science, vol: 4, issue : 1, pp. 91-103, 2016.
- [9] P.D. Wananda, L. Novamizanti, R.D Atmaja, Sistem Deteksi Cacat Kayu dengan Metode Deteksi Tepi SUSAN dan Ekstraksi Ciri Statistik, ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi & Elektronika, Vol 6, No 1, 2018