

# KLASIFIKASI GENRE MUSIK MENGGUNAKAN METODE MEL FREQUENCY CEPSTRUM COEFFICIENTS (MFCC) DAN K-NEAREST NEIGHBORS CLASSIFIER

(Classification of Music Genres Using The Mel-Frequency Cepstrum Coefficients (MFCC) and K-Nearest Neighbors Classifier Methods)

Pandu Deski Prasetyo\*, I Gede Pasek Suta Wijaya, Ario Yudo Husodo

Dept Informatics Engineering, Mataram University  
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: pandudeski@gmail.com, gpsutawijaya@unram.ac.id, ario@unram.ac.id

## Abstract

In the world of music, music has many types of genres such as pop, rock, blues, slow, jazz, metal, dangdut and many more. Everyone has a favorite genre that is different from each other. However, to distinguish it, there is no need to play all music files one by one for many music files. Therefore, computer software is needed to differentiate music genres to make users easier to distinguish the types of music to their wishes automatically. In this paper, a scheme for classifying several types of music genres was developed by using MFCC and KNN methods. The experimental results of this study reached 52.4% with  $k = 13$  as the nearest neighbor parameter.

**Keywords:** MFCC, KNN, Music, Genre, Signal Processing.

\*Penulis Korespondensi

## 1. PENDAHULUAN

Genre merupakan suatu golongan aliran musik yang digolongkan dari kemiripan ritmik, harmoni maupun frekuensi musik kontennya. Genre hal yang penting bagi masyarakat yang mencintai musik terutama dikalangan para anak muda zaman millennial ini karena dapat mudah mengelompokkan genre sesuai dengan yang mereka senangi.

Untuk menggolongkan genre tersebut bisa dengan mendengarkan file musik dengan secara langsung, memang hasil yang diperoleh lebih akurat tapi jika file musik yang di dengar jumlahnya lebih dari 1000 file mungkin akan membuang banyak waktu jika dilakukan secara langsung. Dalam hal ini menggolongkan musik dapat dilakukan dengan cara otomatis walaupun hasil yang diperoleh tidak se akurat mendengarkan langsung [1].

Sebuah file musik harus di ekstraksi terlebih dahulu sebelum dapat di kelompokkan, dengan menggunakan metode MFCC untuk mengekstraksi fitur suatu file musik. MFCC adalah metode yang mengadopsi fungsi kerja telinga manusia yang mampu membedakan suara tinggi dan rendah, biasa digunakan pada *speech recognition* untuk memproses sinyal suara untuk mengetahui ciri dari suara tersebut. Sedangkan

untuk klasifikasi menggunakan KNN yang merupakan algoritma sederhana yang dapat mengategorikan masukkan dengan  $k$  sebagai penentu tetangga terdekatnya [2].

## 2. TINJAUAN PUSTAKA DAN DASAR TEORI

### 2.1. Tinjauan Pustaka

Berikut adalah peninjauan kembali pustaka – pustaka yang berupa penelitian – penelitian sebelumnya tentang masalah yang terkait untuk memberi gambaran metode dan teknik yang digunakan.

Penelitian tentang menentukan ekstraksi ciri untuk pengenalan suara seseorang secara *real-time* [3]. Penelitian ini menghasilkan nilai koefisien MFCC 32 lebih efektif dari *Subband Based Cepstral* (SBC) dari segi tingkat akurasi dan kecepatan proses mengenal suara, baik *text-dependent* maupun *text-independent*. Namun hasil akurasi yang ditunjukkan untuk pengenalan identifikasi suara secara langsung pada MFCC di bawah 70%.

Penelitian tentang proses training dan testing, training untuk prosesnya sedangkan testing untuk klasifikasi nya [4]. Penelitian ini memberikan akurasi terbaik sebesar 67.75% setelah dilakukan pengujian

terhadap sistem dengan beberapa scenario yaitu normalisasi, *codebook* dan *state*, serta jumlah kata.

Penelitian tentang sistem keamanan untuk kunci pintu secara otomatis menggunakan suara. Dengan menggunakan metode MFCC sebagai megekstrasi ciri sinyal suara dan metode DTW sebagai mencocokkan pola suara [5]. Hasil penelitian ini dari beberapa trainer yang mengucapkan kata uji "pintu buka" dan kata uji acak tingkat akuraasi sebesar 81,25%. Tetapi agar dapat menjalankan sistem ini hanya tester yang suaranya terekam di database yang berhasil membuka pintu.

Penelitian tentang studi sistem klasifikasi *genre* pada suatu lagu secara otomatis menggunakan metode naive bayes untuk menarik kesimpulan berdasarkan data yang diperoleh [6]. Data yang diperoleh termasuk data latih dan data tes. Akurasi sistem ini sebesar 83,3% pada 60% data tes.

Penelitian tentang pengenalan suara dirancang untuk membuka kunci pintu dengan mengucapkan kata "buka" dalam keadaan normal, dengan menggunakan metode KNN sebagai pengklasifikasian data dengan cara mencari kerabat dekat dari suara uji berdasarkan tingkat kemiripannya dengan suara-suara latih yang tersedia [7]. Tingkat keberhasilan sistem mengenali suara pengguna didapatkan sebesar 88,88%. Kelemahannya, perbedaan cara pengucapan seperti panjang ucapan, kenyaringan pengucapan kata, intonasi bahkan suara lingkungan sekitar dapat mempengaruhi perbedaan nilai data latih yang didapatkan.

Berdasarkan tinjauan pustaka tersebut, dapat diketahui bahwa pengenalan pola mengenai klasifikasi genre musik pernah dilakukan namun pada penelitian ini sistem pengenalan pola serupa yang membahas tentang genre musik menggunakan metode MFCC dan K-NN sebagai klasifikasi belum pernah dilakukan. Dilihat dari beberapa penelitian yang menggunakan metode MFCC memiliki tingkat akurasi yang beragam. Maka untuk mengetahui seberapa besar tingkat keberhasilan yang dihasilkan oleh metode ini dan di padukan dengan KNN sebagai pengklasifikasi, dilakukan sebuah penelitian klasifikasi genre musik menggunakan metode MFCC dan KNN *classifier*.

## 2.2. Dasar Teori

### 2.2.1. Musik

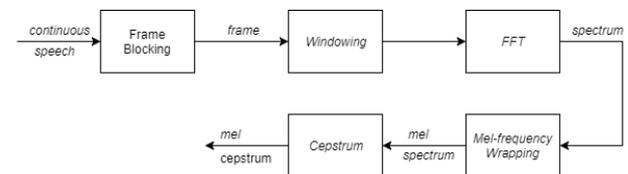
Musik adalah kumpulan suara yang menghasilkan suatu ritme atau alunan yang dapat dinikmati oleh orang banyak. Kata 'musik' berasal dari bahasa Yunani, yaitu *mousike*.

### 2.2.2. Genre Musik

Genre adalah jenis atau aliran pada suatu musik, *genre* pada musik pada dasarnya banyak jenis dan ragam. Ada beberapa negara yang memiliki musik yang memiliki *genre* musik yang mencerminkan budaya negara mereka, salah satunya seperti di Indonesia memiliki musik yang ber-*genre* dangdut yang tidak dimiliki oleh negara lainnya.

### 2.2.3. MFCC

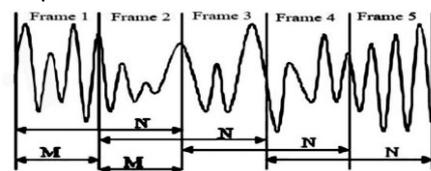
Sinyal suara merupakan sinyal *quasi-stationary*, yaitu ketika diperiksa selama periode yang cukup singkat karakteristiknya cukup stationer. Namun selama jangka waktu yang lama karakteristik sinyal berubah yang mencerminkan perbedaan suara masukan yang diucapkan. Diagram proses MFCC dapat dilihat pada Gambar 1.



Gambar 1. Blok MFCC .

#### a. Frame Blocking

Proses pemisahan suara menjadi beberapa *frame* dan satu *frame* terdiri dari beberapa sampel. Tujuan dilakukan proses ini untuk membentuk sinyal yang tadinya dari *non-stationary* menjadi sinyal yang *quasi-stationary* sehingga dapat diubah dari yang awalnya sinyal domain waktu ke sinyal yang berdomain frekuensi menggunakan transformasi Fourier. Proses ini tampak pada Gambar 2.



Gambar 2. Proses *frame blocking*.

#### b. Windowing

Langkah berikutnya adalah pemrosesan dengan *window* pada masing-masing *frame* individual untuk meminimalisasi sinyal tak kontinu pada awal dan akhir masing-masing *frame*. Konsepnya adalah meruncingkan sinyal ke angka nol pada permulaan dan akhir setiap *frame*.

$$y_1 = x_1(n)\omega(n) \tag{1}$$

dimana:

$y_1$  adalah nilai sampel sinyal *windowing*,

$x_1(n)$  adalah nilai sampel dari *frame* sinyal ke-1,  $\omega$  merupakan fungsi *window*,  $N$  merupakan panjang *frame*.

c. *Fast Fourier Transform (FFT)*

Langkah pemrosesan berikutnya adalah transformasi fourier cepat atau *fast fourier transform*, proses ini mengubah setiap *frame* ke-  $N$  sampel dari domain waktu menjadi domain frekuensi.

$$x_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi jkn}{N}} \quad (2)$$

d. *Mel-frequency Wrapping*

Pendekatan persamaan untuk menghitung *mel* dalam frekuensi  $f$  (Hz) [8] adalah

$$mel(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right) \quad (3)$$

dimana  $f$  = frekuensi sampel.

e. *Cepstrum*

Mengubah spektrum *log mel* menjadi domain waktu. Hasil ini disebut *mel frequency cepstrum coefficient* (MFCC).

$$\tilde{C}_n = \sum_{k=1}^K (\log \tilde{S}_k) \cos \left[ n \left( k - \frac{1}{2} \right) \frac{\pi}{K} \right] \quad (4)$$

dimana:

$\tilde{C}_n$  = koefisien *cepstrum mel – frequency*,

$\tilde{S}_k$  = koefisien daya *mel*.

2.2.4. **Klasifikasi**

Klasifikasi adalah suatu proses untuk mengelompokkan sejumlah data ke dalam kelas-kelas tertentu yang sudah diberikan berdasarkan kemiripan sifat dan pola yang terdapat dalam setiap data. Secara umum, proses klasifikasi dimulai dengan diberikannya sejumlah data latih sebagai acuan untuk membuat aturan klasifikasi suatu data.

Metode KNN adalah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut [2]. KNN merupakan konsep berdasarkan *‘learning by analogy’*. Jika sebuah data *query* yang labelnya tidak diketahui dimasukkan, maka KNN akan mencari  $k$  buah data latih yang jaraknya paling dekat dengan data *query* dalam ruang  $n$ -dimensi. Jarak antara data *query* dengan data latih dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data learning. Jarak antara data latih dan data uji dihitung menggunakan persamaan *Euclidean distance*. Persamaan *distance euclidean* adalah:

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (5)$$

dimana:

$a$  = nilai data latih

$b$  = nilai data uji

Selain *Euclidean distance* ada juga *distance-distance* pada *K-nearest neighbors*, yaitu :

- *City block distance*

*City block distance* disebut juga sebagai *Mahattan distance*. *City block distance* menghitung nilai perbedaan absolut dari dua vektor [9]. Jarak *city-block* didefinisikan dalam persamaan.

$$city(v1, v2) = \sum_{k=1}^N |v1 k - v2 k| \quad (6)$$

Dalam hal ini  $v1$  dan  $v2$  adalah vektor yang akan jaraknya akan dihitung dan  $N$  menyatakan panjang vektor.

- *Cosine distance*

*Cosine distance* adalah fungsi yang menerima dua buah objek dan mengembalikan nilai kemiripan antara kedua buah objek tersebut berupa bilangan riil [10]. *Cosine distance* didefinisikan dalam persamaan.

$$similarity(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \quad (7)$$

dimana :

$x \cdot y$  : vektor *product* dari  $x$  dan  $y$ .

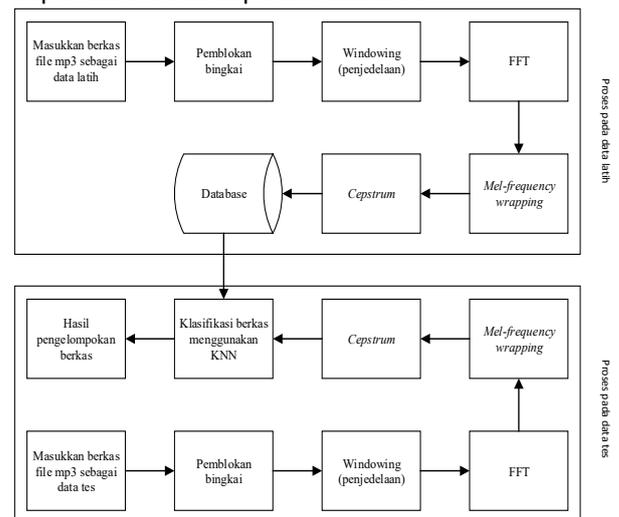
$\|x\|$  : panjang vektor  $x$ .

$\|y\|$  : panjang vektor  $y$ .

3. METODE PENELITIAN

3.1. Gambaran Umum Sistem

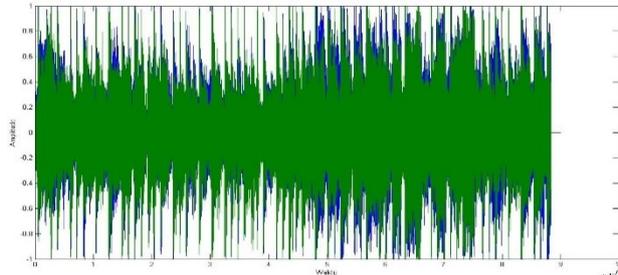
Secara sederhana proses klasifikasi genre musik dapat diilustrasikan pada Gambar 3.



Gambar 3. Gambaran umum sistem.

Gambar 3. menunjukkan proses pengolahan data latih dan pencocokan data tes. Proses mengekstraksi suatu file musik, lalu dikelompokkan sesuai dengan *genre* nya.

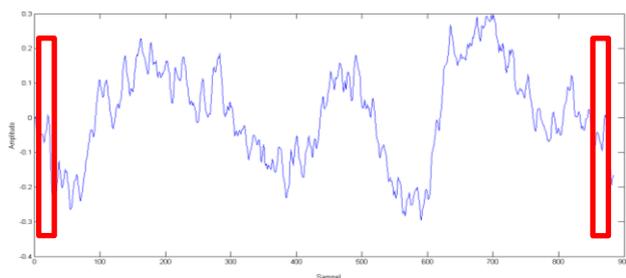
Sebagai contoh kasus, sebuah file musik yang belum diketahui *genrenya* digunakan sebagai *input*. File yang digunakan adalah file yang sudah dilakukan pengeditan berupa potongan 1/3 dari panjang file musik yang di asumsikan bahwa potongan file tersebut menjadi inti dari sebuah *genre* tersebut.



Gambar 4. Contoh data input musik.

Gambar 4. adalah grafik data file musik yang akan diekstraksi fiturnya. Karena file yang digunakan adalah file musik yang memiliki format mp3 maka grafik tersebut terdapat 2 warna yaitu warna hijau dan biru untuk audio yang bersifat 2 *channel*. Langkah per lankah sistem usulan adalah:

**Langkah 1** : *Frame blocking* proses ini membagi data menjadi beberapa sampel. Pada file tersebut frekuensinya 44100 Hz, lalu durasi yang diambil 0,020 (20ms) sehingga untuk mendapatkan titik sampel pada file ini frekuensi dikali panjang durasi yang diambil sehingga didapatkan 882 titik sampel. Setelah titik sampel diketahui, diperoleh grafik pada Gambar 5.



Gambar 5. Grafik data setelah *frame blocking*.

Gambar 5. merupakan hasil data inputan yang telah didapatkan setelah melalui proses *frame blocking*. Sampel pada titik 1 – 5 ditunjukkan oleh Tabel I dan sampel pada titik 876 – 883 ditunjukkan oleh Tabel II. Sampel tersebut ditandai oleh persegi empat berwarna merah. Tabel I. adalah tampilan nilai 5 sampel beserta amplitudo yang terdapat pada awal inputan yang telah dilakukan proses *frame blocking*. Tabel II. adalah tampilan nilai 5 sampel dan amplitudo yang terdapat pada akhir inputan yang telah dilakukan proses *frame blocking*.

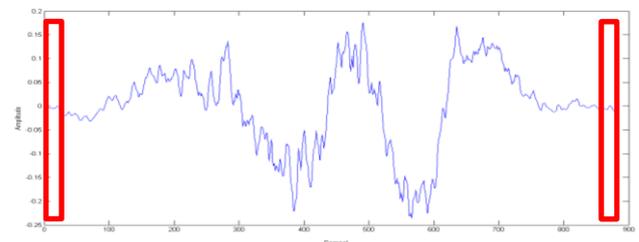
TABEL I. DATA 5 SAMPEL AWAL FRAME BLOCKING

Sampel	Amplitudo (pembulatan 4 angka)
1	0.0039
2	-0.0114
3	-0.0209
4	-0.0318
5	-0.0462
...	...

TABEL II. DATA 5 SAMPEL AKHIR FRAME BLOCKING.

Sampel	Amplitudo (pembulatan 4 angka)
879	-0.1617
880	-0.1864
881	-0.1943
882	-0.1758
883	-0.1658

**Langkah 2** : *Windowing*, setelah hasil *frame blocking* didapatkan maka hasil tersebut melalui proses *windowing* dengan menggunakan persamaan *hamming window*. Maka didapat data sebanyak 883 sampel. Setelah diketahui, diperoleh grafik pada Gambar 6.



Gambar 6. Grafik setelah dilakukan *hamming window*.

Gambar 6. merupakan hasil data inputan yang telah didapatkan setelah melalui proses *hamming window*. Sampel pada titik 1 – 5 ditunjukkan oleh Tabel III dan sampel pada titik 879 – 883 ditunjukkan oleh Tabel IV. Sampel tersebut ditandai oleh persegi empat berwarna merah.

TABEL III. DATA 5 SAMPEL AWAL SETELAH WINDOWING

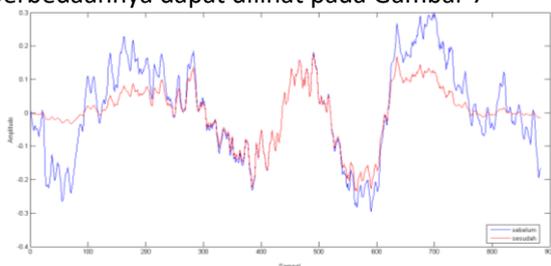
Sampel	Amplitudo (pembulatan 4 angka)
1	0.0003
2	-0.0009
3	-0.0016
4	-0.0025
5	-0.0037
...	...

Tabel III. adalah tampilan nilai 5 sampel dan amplitudo yang terdapat pada awal inputan yang telah dilakukan proses *windowing*.

TABEL IV. DATA 5 SAMPEL AKHIR SETELAH WINDOWING

Sampel	Amplitudo (pembulatan 4 angka)
879	-0.0129
880	-0.0149
881	-0.0155
882	-0.0140
883	-0.0132

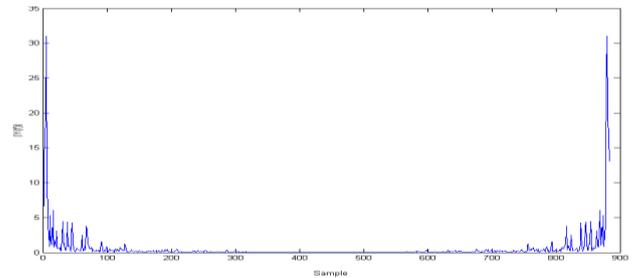
Tabel IV. adalah tampilan nilai 5 sampel dan amplitudo yang terdapat pada akhir inputan yang telah dilakukan proses *windowing*. Dari hasil di atas dapat ditampilkan keseluruhan grafik yang telah melalui proses *windowing*. Diketahui hasil dari *frame blocking* dan hasil dari *windowing* letak perbedaannya dapat dilihat pada Gambar 7



Gambar 7. Perbedaan sebelum dan sesudah *windowing*.

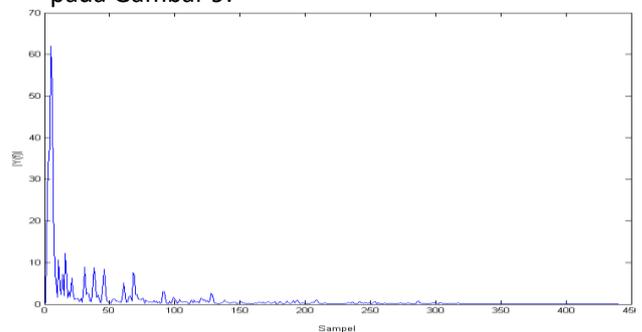
Gambar 7. dapat dilihat adalah perbedaan sinyal data tes sebelum dan setelah melalui proses *windowing*, sinyal yang berwarna biru adalah sinyal yang belum dilakukan proses *windowing* dan sinyal yang berwarna merah menandakan sinyal yang sudah dilakukan proses *windowing*. Letak perbedaann yang signifikan yaitu terdapat pada permulaan dan akhir sinyalnya karena proses *windowing* itu sendiri meminimalisir diskontinuitas sinyal pada permulaan dan akhir setiap *frame* dan konsepnya meruncingkan sinyal ke angka nol pada permulaan dan akhir setiap *frame*.

**Langkah 3:** FFT, pada langkah ini hasil dari frekuensi *hamming window* ini akan melalui proses pada implementasi (2). Hasil FFT merupakan cerminan dari data yang ada,berikut hasil dari implementasi tersebut adalah:



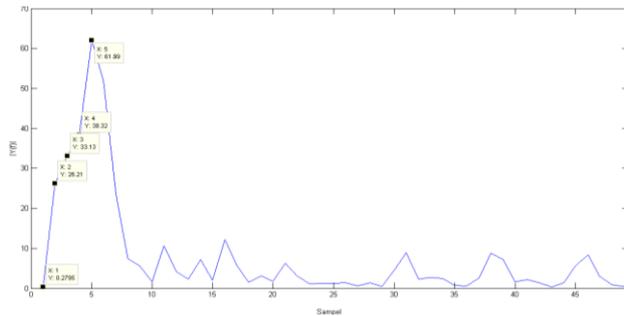
Gambar 8. Frekuensi setelah proses FFT.

Gambar 8. menunjukkan grafik frekuensi yang sudah melalui proses FFT, pada gambar tersebut ditampilkan bentuk data yang tadinya sinyal masih dalam domain waktu sekarang berubah menjadi domain frekuensi atau dapat disebut juga sebagai spektrum. Diterangkan bahwa data pada sisi sebelah kanan adalah cerminan dari data sisi kiri, dimana pada sisi kiri spektrum tersebut adalah sisi positif dan sisi sebelah kiri yang sebagai cerminannya adalah sisi negatif. Sehingga data yang digunakan untuk proses selanjutnya hanya satu sisi yaitu sisi positif yang terletak pada sebelah kiri. Karena keseluruhan terdapat 883 sampel pada FFT dan *frame* yang digunakan hanya sisi positif saja maka jumlah sampelnya pun berubah menjadi 440 sampel dengan kata lain kurang lebih setengah dari banyak *frame* yang digunakan. Berikut adalah grafik frekuensi sisi positif pada hasil FFT dapat dilihat pada Gambar 9.



Gambar 9. Frekuensi *single sided* pada FFT.

**Langkah 4: Mel – Frequency Wrapping**, spektrum yang dihasilkan dari proses FFT selanjutnya diproses menggunakan rumus pada persamaan (3). Proses tersebut menggunakan 100 data sampel sebagai data tesnya, berikut perhitungannya:



Gambar 10. Grafik permulaan pada frekuensi.

Gambar 10. menunjukkan 5 titik sampel yang sudah dilakukan FFT lalu akan dihitung nilai *mel*-nya.

$$mel(0.2795) = 2595 \times \log_{10} \left( 1 + \frac{0.2795}{700} \right) = 0.4499$$

Diatas adalah perhitungan untuk mencari nilai *mel* pada sampel ke 1, sehingga cara yang dilakukan sama sampai data *mel* yang ke 100. Berikut data *mel*-nya:

TABEL V. DATA HASIL MEL.

Sampel	Frekuensi	Mel
1	0.2795	0.4499
2	26.2096	41.4264
3	33.1331	52.1200
4	38.3181	60.0626
5	61.9925	95.6327
...	...	...

Tabel V menampilkan nilai *mel* dari 5 data sampel beserta frekuensi yang terdapat pada masing-masing sampel.

**Langkah 5: Cepstrum**, hasil data dari langkah 4 yang didapat kemudian di ubah kembali dalam bentuk domain waktu dengan menggunakan persamaan (4) yang hasilnya berupa data vektor. Berikut proses perhitungannya:

$$\tilde{C}_1 = \log 0.4499 \cos \left[ 1 \left( 1 - \frac{1}{2} \right) \frac{3.14}{100} \right] = -0.7985$$

TABEL VI. DATA CEPSTRUM

Sampel	Mel	Cepstrum
1	0.4499	-0.7985
2	41.4264	3.7073
3	52.1200	3.88443
4	60.0626	3.7056
5	95.6327	3.4678
...	...	...

Tabel VI. adalah 5 sampel yang sudah diketahui nilai *mel*-nya beserta nilai *cepstrum* yang didapatkan untuk masing-masing nilai *mel*.

**Langkah 6: KNN classifier**, dari hasil ekstra fitur yang didapatkan selanjutnya dilakukan pencocokan data tes yang diperoleh dengan menggunakan klasifikasi K-NN, dalam kasus ini diasumsikan data latih sudah ada maka data tes tersebut dicocokkan dengan data latih yang sudah ada. Dalam kasus ini peneliti menggunakan K=3 maka nilai dari data latih terdekat dan yang medominasi dari nilai vektor data tes tersebut akan menjadi *genre* dari file lagu tersebut. Berikut perhitungannya:

*Cepstrum* data tes:

$$Q = \begin{bmatrix} -0.79 \\ 3.70 \\ 3.84 \\ 3.70 \\ 3.46 \\ \dots \end{bmatrix} \begin{bmatrix} -0.97 \\ -2.50 \\ -0.69 \\ 0.50 \\ 2.30 \\ \dots \end{bmatrix} \begin{bmatrix} 0.97 \\ 1.32 \\ -1.05 \\ -0.47 \\ 0.30 \\ \dots \end{bmatrix}$$

Data *frame* ke-2 dan ke-3 diperoleh dengan langkah yang sama saat mencari data *frame* ke-1.

Didapatkan *cepstrum* file bergenre pop:

$$P1 = \begin{bmatrix} 1.60 \\ 2.52 \\ 2.95 \\ 2.56 \\ 1.65 \\ \dots \end{bmatrix} \begin{bmatrix} -2.62 \\ -1.15 \\ 0.88 \\ 2.36 \\ 2.14 \\ \dots \end{bmatrix} \begin{bmatrix} 0.38 \\ -0.58 \\ -0.99 \\ 0.32 \\ 0.17 \\ \dots \end{bmatrix}$$

$$P2 = \begin{bmatrix} 0.98 \\ 1.55 \\ 3.37 \\ 3.65 \\ 2.72 \\ \dots \end{bmatrix} \begin{bmatrix} -2.70 \\ -1.12 \\ 1.03 \\ 1.95 \\ 1.54 \\ \dots \end{bmatrix} \begin{bmatrix} 0.95 \\ -0.94 \\ -1.62 \\ 0.84 \\ 1.69 \\ \dots \end{bmatrix}$$

Genre rock:

$$R1 = \begin{bmatrix} 1.45 \\ 3.11 \\ 4.20 \\ 4.52 \\ 3.06 \\ \dots \end{bmatrix} \begin{bmatrix} -0.95 \\ -1.30 \\ 1.50 \\ 3.35 \\ 3.51 \\ \dots \end{bmatrix} \begin{bmatrix} 1.26 \\ -2.54 \\ -3.13 \\ 0.85 \\ 1.29 \\ \dots \end{bmatrix}$$

$$R2 = \begin{bmatrix} 0.49 \\ 3.33 \\ 4.56 \\ 4.01 \\ 3.19 \\ \dots \end{bmatrix} \begin{bmatrix} -3.14 \\ -1.35 \\ 1.47 \\ 2.23 \\ 0.55 \\ \dots \end{bmatrix} \begin{bmatrix} 0.86 \\ -1.33 \\ -0.84 \\ 0.34 \\ 1.28 \\ \dots \end{bmatrix}$$

Genre dangdut:

$$D1 = \begin{bmatrix} 1.94 \\ 3.09 \\ 4.03 \\ 1.66 \\ 3.62 \\ dst \end{bmatrix} \begin{bmatrix} -1.88 \\ -0.95 \\ 1.05 \\ 3.24 \\ 3.29 \\ dst \end{bmatrix} \begin{bmatrix} 1.48 \\ -2.00 \\ -1.67 \\ 0.72 \\ 2.05 \\ dst \end{bmatrix}$$

$$D2 = \begin{bmatrix} 2.90 \\ 3.19 \\ 3.44 \\ 3.56 \\ 3.44 \\ dst \end{bmatrix} \begin{bmatrix} -2.93 \\ -1.18 \\ 1.39 \\ 3.50 \\ 2.00 \\ dst \end{bmatrix} \begin{bmatrix} 0.85 \\ -0.61 \\ 1.13 \\ 0.42 \\ 2.20 \\ dst \end{bmatrix}$$

Pada perhitungan ini digunakan inialisasi atau ID dimana:

- Q = data tes,
- P1= file ke-1 bergenre pop,
- P2 = file ke-2 bergenre pop,
- R1 = file ke-1 bergenre rock,
- R2 = file ke-2 bergenre rock,
- D1 = file ke-1 bergenre dangdut,
- D2 = file ke-2 bergenre dangdut.

Data yang yang dicantumkan pada P1,P2,R1,R2,D1,dan D2 adalah data latih yang didapatkan dari perhitungan *cepstrum* yang sama dengan mencari nilai *cepstrum* pada data tes. Data latih yang digunakan pada penelitian ini sebanyak 3 *frame* sebagai sampel yang masing-masing berisi 100 data vektor yang menentukan *genre* pada data tes. Selanjutnya masuk pada perhitungan KNN yang menggunakan *distance euclidean* pada persamaan (2 – 6), berikut perhitungan Q pada data latih P1:

$$d(Q,P1) = \sqrt{\left(\begin{bmatrix} -0.79 \\ 3.70 \\ 3.84 \\ 3.70 \\ 3.46 \\ \dots \end{bmatrix} - \begin{bmatrix} 1.60 \\ 2.52 \\ 2.95 \\ 2.56 \\ 1.65 \\ \dots \end{bmatrix}\right)^2 + \left(\begin{bmatrix} -0.97 \\ -2.50 \\ -0.69 \\ 0.50 \\ 2.30 \\ \dots \end{bmatrix} - \begin{bmatrix} -2.62 \\ -1.15 \\ 0.88 \\ 2.36 \\ 2.14 \\ \dots \end{bmatrix}\right)^2 + \left(\begin{bmatrix} 0.97 \\ 1.32 \\ -1.05 \\ -0.47 \\ 0.30 \\ \dots \end{bmatrix} - \begin{bmatrix} 0.38 \\ -0.58 \\ -0.99 \\ 0.32 \\ 0.17 \\ \dots \end{bmatrix}\right)^2}$$

$$= \sqrt{\begin{bmatrix} -0.95 \\ 5.43 \\ 3.09 \\ 2.19 \\ 5.56 \\ \dots \end{bmatrix}}$$

$$= \sqrt{(-0.95)^2 + (5.43)^2 + (3.09)^2 + (2.19)^2 + (5.56)^2 + \dots}$$

$$= 8.69$$

Didapatkan jarak dari KNN untuk Q,P1 adalah 8.69, dan begitu seterusnya perhitungan ini berlaku pula untuk mencari jarak KNN P2,R1,R2,D1, dan D2. Sehingga didapatkan jarak KNN dari masing-masing ID seperti pada Tabel VII.

TABEL VII. JARAK KNN

ID	Jarak
P1	8.69
P2	8.61
R1	6.73
R2	4.82
D1	5.11
D2	5.94

Tabel VII. adalah hasil *cepstrum* dari file-file musik yang terdiri dari *genre* pop, rock dan dangdut yang telah diberikan ID untuk masing-masing *genre*.

Setelah diketahui jarak dari KNN untuk masing-masing ID, selanjutnya mengurutkan nilai KNN dari nilai yang terkecil ke nilai yang terbesar, sehingga didapatkan:

TABEL VIII. JARAK KNN YANG SETELAH DIURUTKAN KECIL KE BESAR

ID	Nilai
R2	4.82
D1	5.11
D2	5.94
R1	6.73
P2	8.61
P1	8.69

Setelah dilakukan pengurutan jarak dari yang terkecil ke terbesar,karena K=3 maka tinggal mengambil 3 nilai terkecil dari Tabel 8. Sehingga diperoleh:

TABEL IX. DATA 3 JARAK KNN TERKECIL

ID	Jarak
R2	4.82
D1	5.11
D2	5.94

Dilihat dari Tabel 9. ID yang bergenre dangdut lebih mendominasi dari ID yang bergenre rock, maka dapat disimpulkan bahwa klasifikasi *genre* dari data tes tersebut masuk kedalam jenis musik yang bergenre dangdut. Tingkat akurasi yang diperoleh dapat dilihat dari rumus akurasi berikut.

$$Akurasi = \frac{TP+TN}{banyak\ data} \times 100\% \tag{6}$$

dimana:

- TP = true positive,
- TN = true negative.

Pada rumus diatas diketahui bahwa TP adalah jumlah data yang sesuai dengan data latih kemudian dijumlahkan dengan TN dimana nilai TN diketahui dari data yang sesuai dengan suatu *genre*, lalu dibagi dengan keseluruhan data yang digunakan lalu dikalikan dengan 100%. Maka dari rumus tersebut akan diketahui tingkat keakurasian dari setiap *genre* yang ada.

**4. HASIL DAN PEMBAHASAN**

**4.1. Pengujian**

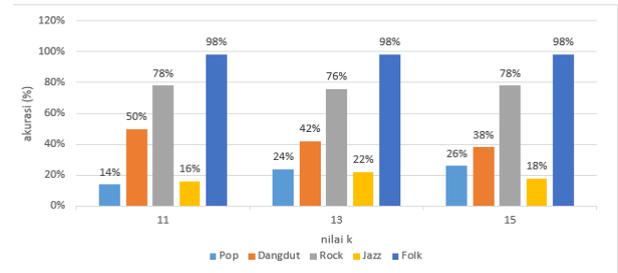
Pengujian pertama dilakukan dengan mengekstraksi data latih yang masing-masing sebanyak 250 *file* musik yang diantaranya 50 *file* lagu bergenre pop, rock, dangdut, jazz, folk yang disimpan ke database, lalu mengekstraksi data testing masing-masing sebanyak 250 *file* dengan jumlah *genre* musik yang sama, serta mencocokkan dengan data latih yang ada pada database dengan menggunakan *Euclidean distance* tujuannya dilakukan pengujian ini untuk mengetahui nilai K terbaik untuk klasifikasi *genre* musik.



Gambar 11. Nilai K optimal.

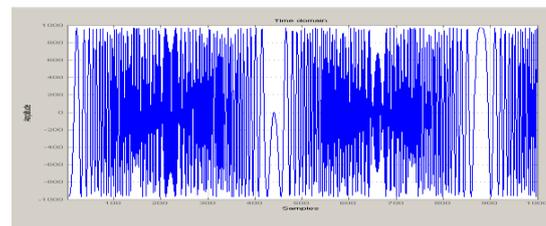
Gambar 11. menggambarkan tingkat akurasi nilai K, jika diperhatikan di peroleh 3 akurasi yang tinggi yaitu nilai K=11, K=13, dan K=15 mempunyai akurasi yang hampir mendekati. Dan didapatkan nilai K=13 sebagai nilai K terbaik karena memiliki tingkat akurasi tertinggi yaitu 52,4%. Dengan waktu komputasi yang dibutuhkan 40,28 menit untuk mencari ekstraksi fitur dari 250 *file* musik pada data testing. Dan dengan rata-rata waktu komputasi yang dibutuhkan 12 detik untuk proses pengklasifikasian.

Pengujian kedua dilakukan dengan jumlah data testing yang sama yaitu 250 *file* musik dengan menampilkan hasil detail dari ketiga nilai K yang didapat dari pengujian pertama untuk masing-masing *genre*.



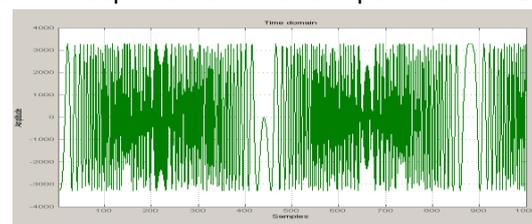
Gambar 12. Akurasi *genre* nilai K

Dilihat pada Gambar 12. nilai K = 11, K = 13 dan K = 15 adalah hasil dari setiap *genre* untuk masing-masing nilai K. Musik yang memiliki *genre* rock dan folk akurasi yang di dihasilkan rata-rata lebih dari 70% dibanding musik yang bergenre pop, dangdut maupun jazz yang keakuratannya dibawah 50%. Hal ini disebabkan saat proses klasifikasi dimana waktu proses menggunakan *Euclidean distance* untuk mencari jarak yang terdekat, kemungkinan jarak yang terdekat merupakan jenis *genre* yang sama namun selain melihat jarak yang terdekat kemungkinan juga dilihat dari jumlah jenis *genre* dominan yang muncul walaupun jarak antara data tes tidak terlalu dekat, maka hal ini juga bisa berpengaruh terhadap nilai K yang digunakan untuk menentukan hasil dari klasifikasi dari suatu *genre* tersebut. Sebab tersebut dapat dilihat pada Gambar 13.

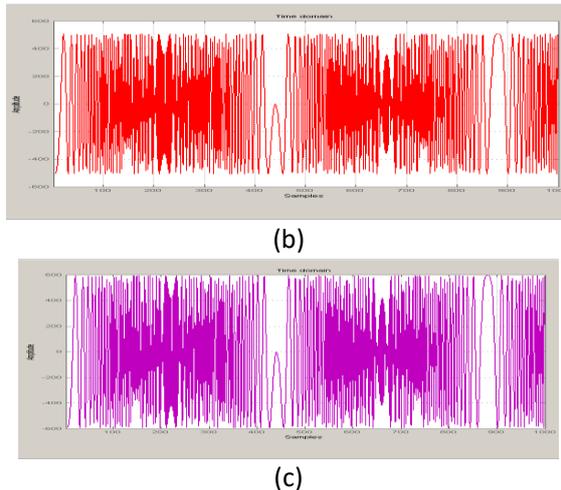


Gambar 13. Sebaran data dari fitur MFCC lagu folk.

Gambar 13. merupakan plot dari fitur musik yang bergenre folk, dimana plot ini salah satu patokan dari database untuk musik yang masuk dalam *genre* folk dari database. Namun ada beberapa *genre* musik lain yang masuk kedalam *genre* folk, salah satunya ada musik yang bergenre pop, dangdut dan jazz. Setelah dilihat plot dari salah satu file yang masuk kedalam *genre* folk dapat dilihat kesamaan pada Gambar 14.



(a)



Gambar 14. Sebaran data dari fitur MFCC lagu (a) pop (b) dangdut dan (c) jazz.

Gambar 4.4 menunjukkan plot dari fitur musik yang bergenre pop, dangdut dan jazz. Dari plot ekstraksi fitur diatas dapat diketahui bahwa plot hasil ekstraksi fitur yang dihasilkan secara visual terlihat sama namun jika dari segi nilai statistik akan di korelasikan dengan musik bergenre folk pada database. Diketahui perbandingan sebaran lagu pop terhadap sebaran lagu folk pada database memiliki korelasi sebesar 0.7122, sebaran lagu dangdut terhadap lagu folk memiliki korelasi sebesar 0.8684, dan sebaran lagu jazz terhadap lagu folk memiliki korelasi sebesar 0.8965. Suatu fitur bisa dikatakan mirip jika korelasi antara kedua vektor tersebut bernilai 1 atau mendekati 1, sedangkan jika mendekati nilai 0 dikatakan tidak mirip. Karena plot yang dihasilkan dari ketiga sebaran mendekati nilai 1 dapat disimpulkan ketiga *genre* memiliki kemiripan yang tinggi, sehingga lagu yang bergenre pop, dangdut dan jazz dikenali sebagai musik yang bergenre folk.

## 5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian pada bab sebelumnya, maka dapat disimpulkan bahwa:

1. Nilai K terbaik dari metode kNN untuk penelitian ini adalah 13 dengan pencapaian tingkat akurasi sebesar 52,4%.
2. *Genre* pop, dangdut, dan jazz rata-rata tingkat akurasi nya untuk nilai K=5 sampai K=35 mendapatkan persentase dibawah 50%.

3. Nilai K sangat mempengaruhi hasil proses dari klasifikasi.

Karena hasil klasifikasi belum maksimal maka diperlukan suatu sistem yang berbasis *deep-learning* seperti *Recurrent Neural network* dengan data sample yang lebih banyak.

## DAFTAR PUSTAKA

- [1] L. Arida, A. Rahning, and S. Hartati, "Klasifikasi Genre Musik Menggunakan Learning Vector Quantization dan Self Organizing Map," vol. 9, no. 1, pp. 14–22, 2016.
- [2] Brilliant.org, "K-nearest Neighbors." [Online]. Available: <https://brilliant.org/wiki/knearest-neighbors/>. [Accessed: 09-Jul-2018].
- [3] P. Prasetyawan, "Perbandingan Identifikasi Pembicara Menggunakan Mfcc Dan Sbc Dengan Ciri Pencocokan Lbg-Vq," vol. 2016, no. Sentika, pp. 18–19, 2016.
- [4] R. M. Fauzi and W. Maharani, "Pengenalan Ucapan Huruf Hijaiyah Menggunakan Mel Frequency Cepstral Coefficients (MFCC) Dan Hidden Markov Model (HMM)," *Telkom Univ.*, 2013.
- [5] Z. Effendi, T. Erlina, and R. Aisuwarya, "Pengenalan Suara Menggunakan Metode MFCC ( Mel Frequency Cepstrum Coefficients ) dan DTW ( Dynamic Time Warping ) untuk Sistem Penguncian Pintu ISBN : 979-26-0280-1," *Semin. Nas. Teknol. Inf. dan Komun. Terap. 2015*, pp. 239–243, 2015.
- [6] A. Priagung, "Klasifikasi Genre Pada Lagu Dengan Metode Naive Bayes," Universitas Brawijaya, 2012.
- [7] A. Apriyansyah, "Prototype Kunci Otomatis Pada Pintu Berdasarkan Suara Pengguna Menggunakan Metode Knn (K-Nearest Neighbor)," *Coding*, vol. 04, no. No.1, pp. 45–56, 2016.
- [8] A. Mustofa, "Sistem Pengenalan Penutur dengan Metode Mel-frequency Wrapping," *J. Tek. Elektro*, vol. 7, no. 2, pp. 88–96, 2007.
- [9] K. Riset *et al.*, "Implementasi Metode City Block Distance," 2017.
- [10] A. Triana and F. Mipa, "Pemanfaatan Metode Vector Space Model dan Metode Cosine Similarity pada Fitur Deteksi Hama dan Penyakit Tanaman Padi," pp. 1–6.