

KLASIFIKASI MOOD MUSIK MENGGUNAKAN K-NEAREST NEIGHBOR DENGAN MEL FREQUENCY CEPSTRAL COEFFICIENTS

(Mood Music Classification using K-Nearest Neighbor with Mel Frequency Cepstral Coefficients)

Fuad Fadlila Surenggana, Arik Aranta*, Fitri Bimantoro

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: fuad.surenggana@gmail.com, [arikaranta, bimo]@unram.ac.id

Abstract

Music is a combination of tones, vocals and also instrumental harmony to express something that is emotional. Mood music can affect human emotions. Musical moods can increase arousal and pleasure and can affect communication emotions. Music can be found easily through the Digital Music Library (DML) but the grouping of music in DML is limited to title, singer, album and genre. There needs to be variations in recognizing music to reach a wider grouping and classification of music. This study aims to classify musical moods based on Mel-Frequency Cepstral Coefficients (MFCC) and using K-Nearest Neighbor (KNN). The dataset used is 200 music files and is divided into 4 mood classes based on Thayer's model, namely angry (angry), happy (happy), sad (sad), and relax (relaxed). Feature extraction using MFCC will produce 13 MFCC features, 13 delta features and 13 delta-delta features. In the study, the accuracy was 85.5% using KNN with a value of $k=5$ and using the Manhattan distance method.

Keywords: Klasifikasi, Mood, Musik, K-Nearest Neighbor, Mel-Frequency Cepstral Coefficients

*Penulis Korespondensi

1. PENDAHULUAN

Musik merupakan kombinasi antara nada, vokal, dan juga instrumental yang harmoni untuk mengekspresikan sesuatu yang bersifat emosional. Menurut Meng[1], musik berpengaruh secara signifikan terhadap emosi, *mood* musik yang terdengar menyenangkan dan menggetarkan dapat meningkatkan gairah, tempo pada musik juga memiliki efek yang signifikan terhadap emosi komunikasi, tempo musik yang lebih cepat dapat meningkatkan gairah dan kesenangan. Saat ini, musik dapat ditemukan dengan sangat mudah terutama melalui *Digital Music Library* (DML). Namun demikian, secara umum, pengelompokan musik pada DML terbatas pada judul, nama penyanyi, album, dan genre musik. Hal ini berakibat pada kurangnya pengelompokan dan klasifikasi musik sesuai dengan kebutuhan pendengar. Dengan klasifikasi *mood* musik yang tepat, musik dapat digunakan untuk membangun emosi di tempat tertentu dan pada situasi tertentu.

Berbagai metode untuk mengklasifikasi mood musik pernah digunakan pada penelitian sebelumnya menggunakan *Backpropagation Neural Network*

(BPNN) dan *Mel Frequency Cepstral Coefficient* (MFCC) dengan menggunakan 200 dataset berhasil mendapatkan akurasi sebesar 87,67%[2], penelitian lain menggunakan metode *Artificial Neural Network* (ANN) mendapatkan akurasi 73%[3], kombinasi hierarkis *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN)[4] mendapat akurasi 84%. *Neural Network* (NN) memiliki kelebihan pada prediksi non-linear, kuat pada *parallel processing* dan kemampuan untuk mentoleransi kesalahan, tapi memiliki kelemahan pada perlunya data *training* yang besar, *overfitting*, lambatnya konvergensi, dan sifatnya yang local optimum. Penelitian-penelitian tersebut menggunakan metode yang berbeda-beda mulai dari metode tradisional seperti SVM dan KNN sampai metode yang sedang berkembang yaitu BPNN dan ANN, namun dari segi akurasi dapat terlihat memiliki akurasi yang tidak terlalu jauh signifikan.

K-Nearest Neighbor (KNN) merupakan metode *supervised learning*. *Instance* baru diklasifikasikan berdasarkan sampel *training* terdekat di ruang fitur[6]. KNN dapat menyelesaikan masalah SVM dan NN, yaitu lebih efektif pada data yang berdimensi rendah dan tidak membutuhkan data yang terlalu

besar. Pada penelitian ini kami mengusulkan menggunakan KNN dengan melihat fakta di atas bahwa akurasi dari penelitian-penelitian sebelumnya tidak jauh berbeda antara satu dengan lainnya.

Mel Frequency Cepstral Coefficient (MFCC) adalah suatu metode yang digunakan untuk mengekstraksi fitur dari sinyal audio. Metode ini merupakan metode yang paling banyak digunakan di berbagai bidang pemrosesan suara, pendekatan terdepan dalam teknik ekstraksi fitur ucapan, dan teknik yang cukup baik untuk menghasilkan fitur untuk digunakan pada klasifikasi *mood*[7][8][3]. MFCC didasarkan pada variasi yang diketahui dari *bandwidth* kritis terhadap frekuensi pada telinga manusia[5].

Berdasarkan uraian di atas, penulis ingin membuktikan bahwa hasil akurasi KNN yang memiliki komputasi lebih sederhana dengan memori yang lebih kecil dan lebih mudah diterapkan dapat mendekati hasil akurasi BPNN yang memiliki komputasi yang kompleks dan memori yang lebih besar. Penelitian ini untuk membuat model untuk melakukan klasifikasi *mood* musik menggunakan *K-Nearest Neighbor* dan *Mel Frequency Cepstral Coefficients*. Dataset memiliki data yang sama seperti penelitian sebelumnya[2] menggunakan 200 data dengan *mood contentment*, *exuberance*, *depression*, dan *anxious*. Penelitian ini bertujuan menguji penggabungan metode KNN dan MFCC dalam performa untuk mengelompokkan musik ke dalam empat kelompok *mood* atau suasana hati model Thayer.

2. TINJAUAN PUSTAKA

Pada penelitian sejenis, yaitu klasifikasi *mood* musik, pernah dilakukan menggunakan MFCC dan BPNN dengan hasil akurasi sebesar 87,67%. Penelitian ini dilakukan dengan cara membagi *mood* menjadi 4 dengan model Thayer masing masing mood 50 data dan durasi data audio 60 detik. MFCC digunakan untuk ekstraksi fitur dengan pengaturan panjang *frame* 40ms, dan *overlap* 40%[2].

Penelitian lain klasifikasi musik berdasarkan genre dan *mood*, peneliti menggunakan 600 data dibagi menjadi 6 kelas untuk ekstraksi fitur yang mendapatkan MFCC sebagai fitur terbaik untuk klasifikasi *mood*. Proses klasifikasi, menggunakan SVM dan ANN dengan Tingkat akurasi SVM adalah 70% dan ANN sebesar 73% untuk klasifikasi *low/high arousal* dan keduanya mendapat akurasi sebesar 63% untuk *low/high valence* pada model diagram Thayer[3].

Penelitian klasifikasi emosi pada musik menggunakan hierarkis SVM dan KNN. Pada penelitian ini, peneliti menggunakan 120 data untuk 4

kelas, ekstraksi fitur berdasarkan 4 grup fitur (*Energy*, *Rhythm*, *Pitch* dan *Timbre*) yang menghasilkan total 127 fitur ekstraksi. Klasifikasi menggunakan kombinasi SVM dan KNN secara hierarki menjadi dua level. Level pertama *dataset* diklasifikasi menggunakan SVM menjadi *high energy* dan *low energy* (klasifikasi model Thayer) dan level kedua klasifikasi menggunakan KNN dengan nilai $k=3$ menjadi ruang positif dan negatif. Hasilnya mendapatkan akurasi sebesar 84%[4].

Penelitian mengenai pengelompokan musik untuk menentukan suasana hati dilakukan menggunakan metode *Fast Fourier Transform* untuk mentransformasi sinyal menjadi domain frekuensi dan *spectral analysis* untuk mengekstraksi fitur menghasilkan 9 fitur. *Dataset* berjumlah 400 data terbagi menjadi 4 kelas diklasifikasi menggunakan KNN dengan akurasi 86,55% dengan nilai $k=5$ [9].

Pada penelitian yang menggunakan KNN dan MFCC meneliti tentang ECG sinyal yang menggunakan MFCC sebagai ekstraksi fitur menghasilkan 13 fitur dan klasifikasi KNN dengan nilai $k=3$ dan menggunakan *Euclidean distance*. Pengujian menggunakan *K-fold Cross validation* dan *performance matrix*. *Dataset* berjumlah 100 data untuk 2 kelas. Hasil penelitian mendapat tingkat *accuracy* 84%, *sensitivity* 85% and *specificity* 84%[10].

Penelitian mengenai klasifikasi deteksi kesalahan pada bantalan pada rotasi mesin. Ekstraksi fitur menggunakan MFCC dan Delta MFCC dan klasifikasi menggunakan *Convolution Neural Network* dengan hasil akurasi sebesar 98,76%[11]. Pada penelitian sistem pembuka pintu otomatis, MFCC dan DTW digunakan untuk memproses fitur suara dari pengguna. didapatkan akurasi sebesar 81,25%[12]. Penelitian pengenalan suara sebagai pengontrol lampu otomatis menggunakan MFCC dan *Adaptive Neuro-Fuzzy Inferense System* (ANFIS) dengan 20 data dengan masing-masing data menghasilkan 6 fitur, hasil uji akurasi untuk data suara terlatih sebesar 98,57 % dan data tak terlatih sebesar 95,90%[13].

Berdasarkan tinjauan pustaka tersebut di atas penulis mengajukan penelitian dengan menggunakan MFCC dan klasifikasi KNN. Adapun pertimbangan menggunakan MFCC karena MFCC memiliki hasil terbaik untuk penelitian mengenai pengenalan suara. Begitu juga dengan KNN yang memiliki hasil yang cukup baik untuk metode yang mudah diimplementasikan dan relatif lebih cepat untuk pemrosesannya dibanding NN karena tidak perlu membuat *training* model.

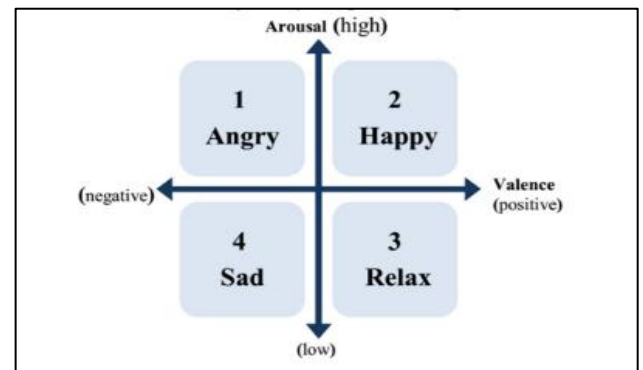
2.1. Musik

Musik adalah suara yang dirasakan seseorang dan bergantung pada sejarah, lokasi, budaya, dan selera pribadi. Definisi musik yang sebenarnya juga beragam. Misalnya, (1) musik adalah bunyi/kesan sesuatu yang ditangkap oleh pancaindra pendengar, (2) musik adalah karya seni dengan segala unsur mayor dan minor, dan (3) musik adalah bunyi yang sengaja dihasilkan oleh seseorang. atau sekelompok orang yang disajikan sebagai musik. Dari beberapa definisi tersebut, musik adalah suara yang sengaja dihasilkan oleh manusia yang disajikan sebagai musik. Musik memiliki tiga bagian penting: bit, ritme, dan harmoni. Musik yang baik adalah musik yang menyeimbangkan ketiganya menjadi satu kesatuan yang harmonis. Ketiga bagian penting tersebut masing-masing memiliki dampak yang berbeda bagi kehidupan manusia. Irama mempengaruhi jiwa, dan melodi mempengaruhi pikiran dan jiwa, seperti halnya sedikit mempengaruhi tubuh[14].

2.2. Suasana Hati (*Mood*) pada Musik

Mood atau suasana hati adalah keadaan emosi yang relatif bertahan lama. Suasana hati berbeda dari emosi sederhana dalam hal mereka kurang spesifik, kurang intens, dan kurang mungkin dipicu oleh stimulus atau peristiwa tertentu. Mendengarkan sebuah lagu dapat merangsang suasana hati pendengarnya tergantung dari musik yang didengarkannya[15].

Berdasarkan penelitian James Russell, Robert Thayer mengusulkan model suasana hati dua dimensi pada tahun 1989. Model yang diusulkan menyediakan cara yang lebih sederhana dan lebih efektif untuk mengekspresikan suasana hati[14]. Model Thayer menyatakan suasana hati tergantung pada dua faktor, yaitu: *stress* (*happines/kebahagiaan* dan *anxiety/kecemasan*), dan energi (*calm/tenang* dan *energy/berenergi*) dikombinasikan dalam sumbu dua dimensi yang membentuk empat kuadran yang berbeda yaitu: *contentment* mewakili jenis *mood* musik yang tenang dan bahagia, *depression* mewakili jenis *mood* musik yang galau, lirih dan depresi, *exuberance* mewakili jenis *mood* musik yang mengacu pada bahagia dan energik dan *anxiety* mewakili jenis musik yang panik, cemas, dan kalut[16]. Model *mood* Thayer dapat dilihat pada Gambar 1.



Gambar 1. Model dimensi emosi Thayer[16]

Tidak terdapat basis data standar untuk mengklasifikasi *mood* musik sehingga kebanyakan peneliti menggunakan data mereka masing-masing. Hal ini dikarenakan belum terdapat konsensus mengenai model emosi maupun jumlah emosi yang harus diklasifikasi[27].

Dataset yang digunakan dalam penelitian ini adalah *file* musik instrumental dengan empat model kategori *mood* Thayer yaitu *contentment* untuk *mood relax*(santai), *depression* untuk *mood sad*(sedih), *exuberance* untuk *mood Happy*(Bahagia), dan *anxiety* untuk *mood Angry*(marah). Model Thayer digunakan karena kesederhanaannya mengategorikan emosi menjadi dua dimensi yaitu *arousal* dan *valence* dan kemudahan relatif yang dapat diintegrasikan dengan variabel dari studi penggunaannya dan fitur audio.

Kelemahan dari pendekatan ini adalah dua dimensi tersebut tidak cukup untuk menjelaskan semua perbedaan dalam emosi yang diekspresikan oleh musik karena kesederhanaan dimensinya. Namun demikian, kesederhanaan modelnya dan tidak terdapat korelasi signifikan antara fitur musik di luar *arousal* dan *valence*[28], membuat model Thayer digunakan sebagai dasar melakukan klasifikasi *mood*. Oleh karena model Thayer data dikatakan lebih efektif untuk mewakili suasana hati dibandingkan model Russell, maka pada penelitian ini digunakan model Thayer sebagai dasar melakukan klasifikasi *mood*.

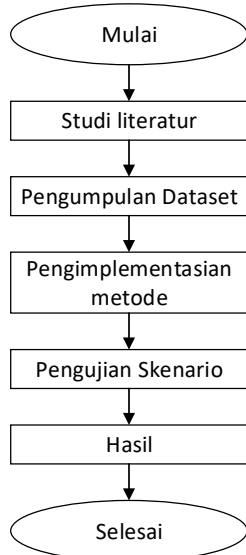
3. METODE PENELITIAN

3.1. Alur Penelitian

Terdapat beberapa tahapan dalam pelaksanaan penelitian ini. Pertama dilakukan studi literatur untuk memperoleh wawasan atau dasar dalam melakukan penelitian. Studi literatur dilakukan dengan membaca jurnal, buku, dan penelitian-penelitian terkait pemrosesan suara.

Dataset yang digunakan berdasarkan data penelitian sebelumnya[2] yang dikumpulkan dari situs

epidemicsound.com. Setiap *mood* memiliki 50 data menjadikan total keseluruhan *dataset* berjumlah 200 *dataset*. Pada situs tersebut terdapat berbagai filter berbagai jenis musik untuk dipilih, namun pada penelitian ini hanya mengambil empat jenis *mood* sesuai dengan model Thayer[2].



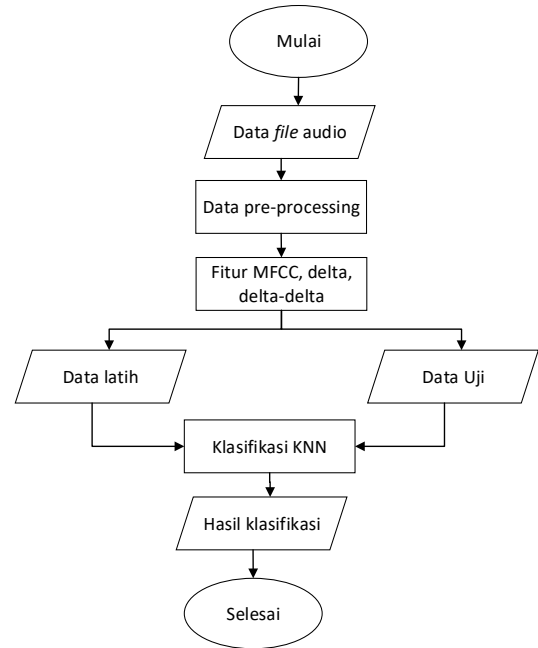
Gambar 2. Diagram alur penelitian

. *Mel Frequency Cepstral Coefficient* (MFCC) digunakan untuk mengekstraksi fitur sinyal audio. Terdapat masing-masing 13 koefisien MFCC, delta dan delta-delta yang akan diekstrak untuk setiap file dengan durasi 60s, Panjang *frame* 40ms dan *overlap* 40%[2]. Kemudian, dilakukan Klasifikasi menggunakan *K-Nearest Neighbor* (KNN). Pada proses ini dilakukan manipulasi pada variabel skenario uji antara lain variasi jarak yang dibutuhkan pada perhitungan tetangga KNN dan perhitungan nilai k terbaik

Setelah pengujian skenario selesai, maka dilakukan analisis hasil pengujian skenario menggunakan *k-fold cross validation* dan *confusion matrix* yang selanjutnya diambil jarak dan nilai k dengan akurasi terbaik. Diagram alir penelitian dapat dilihat pada Gambar 2.

3.2. Perancangan Sistem

Dalam pengimplementasian *machine learning* menggunakan ekstraksi fitur MFCC dengan klasifikasi KNN, terdapat berbagai tahapan yang dilakukan, yaitu, ekstraksi fitur, pembagian data latih dan uji, dan proses klasifikasi KNN. Diagram alir pengimplementasian metode dapat dilihat pada Gambar 3.2.



Gambar 3. Diagram alur perancangan sistem

3.2.1. Data Pre-processing

Proses *pre-processing* yang dilakukan yaitu dengan mengubah bentuk *file .wav* ke bentuk sinyal amplitudo dengan frekuensi 44100Hz dan memotong sinyal menjadi berdurasi 60 detik. Sinyal yang diinginkan yaitu pada bagian tengah sinyal karena pada durasi dan bagian lagu tersebut dapat mencakup *verse* dan *chorus* dari suatu lagu

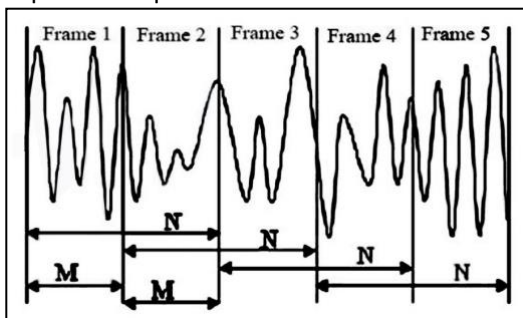
3.2.2. Ekstraksi Fitur MFCC

Ekstraksi fitur MFCC untuk mendapatkan fitur dari setiap *file* audio. Fitur yang akan diekstraksi terdiri dari 13 koefisien MFCC, delta dan delta-delta (delta dua) untuk setiap *file* audio. Terdapat beberapa parameter yang digunakan pada proses ekstraksi fitur MFCC yaitu dengan durasi 60 detik, panjang *frame* 40 dan *overlap* sebesar 40%[2]. Dan juga menggunakan *hamming window* sebagai *windowing* pada salah satu tahapan pada MFCC. Setelah koefisien-koefisien MFCC diperoleh, selanjutnya melakukan *scaling* pada data-data tersebut sebagai salah satu *scenario* uji. *Scaling* yang digunakan yaitu terdiri dari, *Min-Max scaling*, *Standard scaling* dan *Robust scaling*. Adapun tahapan MFCC adalah sebagai berikut:

a. Frame Blocking

Proses ini melakukan segmentasi sinyal dalam *frame* menjadi bagian yang lebih kecil. Ukuran dari *frame* harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik. Tetapi ukuran *frame* juga harus cukup pendek untuk dapat menunjukkan resolusi

waktu yang baik. Sinyal suara dibagi menjadi *frame* N sampel. *Frame* yang berdekatan dipisahkan oleh M ($M < N$)[18]. N merupakan Panjang *frame* dan M merupakan titik dimulainya untuk *frame* selanjutnya sehingga menghasilkan kumpulan *frame* yang overlap. Panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30-50 persen dari panjang *frame*. *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame*[19]. Contoh ilustrasi *frame blocking* dapat dilihat pada Gambar 4



Gambar 4. Contoh *frame blocking*[2]

b. *Windowing*

Sinyal suara yang telah melalui *frame blocking* akan menyebabkan kesalahan pada proses *fourier transform*. *Windowing* bertujuan untuk mengurangi terjadinya kebocoran *spectral* atau *aliasing* yang merupakan suatu efek dari timbulnya sinyal baru yang memiliki frekuensi yang berbeda dengan sinyal aslinya. Efek tersebut dapat terjadi karena rendahnya *sampling rate* atau karena proses *framing* yang menyebabkan sinyal menjadi diskontinu. *Windowing* dilakukan pada sinyal diskrit[19]. Representasi fungsi *windowing* menggunakan persamaan (1)

$$X(n) = f_l(n)w(n), 0 \leq n \leq N - 1 \quad (1)$$

dengan, fungsi $X(n)$ adalah signal hasil *windowing*, f_l adalah hasil *frame blocking*, dengan n adalah $0,1,\dots,N-1$. N adalah jumlah sampel dalam masing-masing *frame* dan $w(n)$ adalah fungsi *window*.

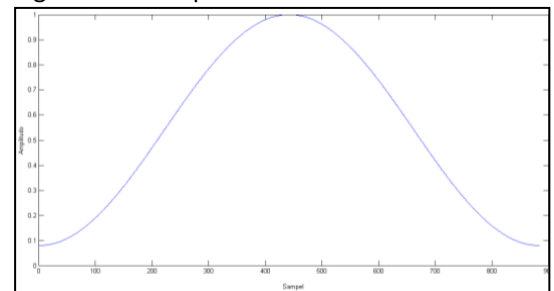
Jenis *windowing* yang digunakan yaitu *Hamming Window*. *Hamming Window* digunakan sebagai bentuk *window* dengan mempertimbangkan blok berikutnya dalam rantai pengolahan ekstraksi fitur dan

mengintegrasikan semua garis frekuensi yang terdekat[19]. *Hamming window* dapat diperoleh dengan persamaan (2).

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

dengan, $w(n)$ adalah fungsi *window* menggunakan *hamming*, dengan n adalah bilangan bulat $0,1,\dots,M-1$, M adalah panjang *frame*.

Dalam bentuk grafik *hamming window* digambarkan seperti Gambar 5.



Gambar 5. Ilustrasi *hamming window*[2]

c. *Fast Fourier Transform*

Fast Fourier Transform (FFT) adalah suatu algoritma yang digunakan untuk mengonversi setiap *frame* N sampel dari domain waktu ke domain frekuensi. FFT adalah algoritma cepat untuk mengimplementasikan discrete fourier transform (DFT) dengan didefinisikan melalui Persamaan (3):

$$d(k) = \sum_{n=0}^{N-1} X(n) e^{-\frac{2\pi jkn}{N}}, 0 \leq n \leq N - 1 \quad (3)$$

dengan, $d(k)$ adalah hasil perhitungan DFT, $X(n)$ adalah hasil *windowing*, N adalah jumlah sampel yang akan diproses pada masing-masing *frame*, k adalah variable frekuensi diskrit $0,1,2,\dots,(N-1)$, j adalah bilangan imajiner ($\sqrt{-1}$), dan $n=1,2,3,\dots,(N-1)$.

FFT mempunyai tujuan mendekomposisi sinyal menjadi sinyal sinusioda yang berupa unit real dan unit imajiner. FFT menggunakan persamaan (4)

$$T(k) = \sum_{n=0}^{N-1} X(n) \cos\left(\frac{2\pi kn}{N}\right) - \sum_{n=0}^{N-1} X(n) \sin\left(\frac{2\pi kn}{N}\right) \quad (4)$$

Dengan, fungsi $T(k)$ adalah hasil perhitungan FFT ke- k , $X(n)$ adalah hasil perhitungan windowing ke- n , n adalah nomor urut sinyal dan k adalah indeks dari frekuensi $(1,2,...N)$.

d. *Mel-Frequency Wrapping*

Persepsi sistem pendengaran manusia terhadap frekuensi sinyal suara tidak dapat diukur dalam skala linear. Untuk setiap nada dengan frekuensi aktual, f , diukur dalam Hz , sebuah *subjective pitch* diukur dalam sebuah skala yang disebut *Mel*. Skala *mel-frequency* ialah sebuah frekuensi rendah yang bersifat linear di bawah 1000Hz dan sebuah frekuensi tinggi yang bersifat logaritmik di atas 1000Hz[19]. MFW mempunyai tujuan menghasilkan spektrum mel menggunakan Persamaan (5)

$$Y(i) = \sum_{j=1}^G T(j) H_i(j) \quad (5)$$

dengan, $Y[i]$ adalah hasil perhitungan *mel frequency wrapping* ke- i , G adalah jumlah *magnitude spectrum* ($G \in \mathbb{N}$), $T[j]$ adalah hasil FFT, $H_i[j]$ adalah *coefficient filterbank* pada frekuensi j ($1 \leq i \leq E$), dan E adalah jumlah *channel* dalam *filterbank*.

Persamaan berikut untuk melakukan konversi frekuensi biasa (linear) ke frekuensi Mel (6) dan sebaliknya (7) dapat dilihat sebagai berikut :

$$mel(f) = 1125 \times \ln\left(1 + \frac{f}{700}\right) \quad (6)$$

$$mel^{-1}(mel) = 700\left(\exp\left(\frac{mel}{1125}\right) - 1\right) \quad (7)$$

dengan, $mel(f)$ adalah hasil perhitungan frekuensi *mel*, f adalah frekuensi, mel^{-1} adalah invers fungsi *mel*.

Proses *mel-frequency wrapping* di mulai dengan menentukan jumlah filter yang akan digunakan. Jumlah filter yang digunakan biasanya berkisar antara nilai 20 hingga 40[20].

e. *Cepstrum*

Langkah terakhir ekstraksi ciri dari proses ekstraksi ciri adalah mengubah domain hasil dari *filterbank* yang masih dalam domain frekuensi ke domain waktu menggunakan

perumusan *Discrete Cosine Transform* (DCT) untuk memperoleh nilai koefisien MFCC-nya. Pada tahap terakhir ini kita dapat menentukan jumlah koefisien yang akan digunakan sebagai ciri. Jumlah koefisien i minimum yang umumnya diambil untuk proses pengenalan suara adalah 12-20 atau maksimum sejumlah filter yang didefinisikan[20]. Perumusan untuk perolehan nilai koefisien MFCC ditunjukkan pada Persamaan (8).

$$C(m) = \sum_{k=1}^K \log_{10} Y(k) \cos \left[m\left(k - \frac{1}{2}\right) \frac{\pi}{K} \right] \quad (8)$$

dengan, C_j adalah *coefficient*, $Y(k)$ adalah keluaran dari proses *filterbank*, m adalah banyaknya *coefficient* $1,2,...,K$, K adalah jumlah *coefficient* yang diharapkan.

f. *Delta Cepstrum*

Delta cepstrum mencerminkan variasi MFCC, ini menunjukkan karakteristik dinamis[11]. *Delta cepstrum* dari fitur N dimensi MFCC ditunjukkan pada persamaan (9).

$$d(n) = \begin{cases} C(n+1) - C(n) & n < K \\ \frac{\sum_{k=1}^K k[C(n+k) - C(n-k)]}{\sqrt{2 \sum_{k=1}^K k^2}} & \text{others} \\ C(n) - C(n-1) & n \geq N - K \end{cases} \quad (9)$$

dengan, $d(n)$ adalah nilai ke- n *delta cepstrum*, C adalah hasil perhitungan cepstrum dan K adalah perbedaan waktu, biasanya 1 atau 2.

3.2.3. *Data Scaling*

a. *Min-Max Scaling*

Normalisasi *min-max* mengubah ukuran data dari rentang asli, sehingga semua nilai berada dalam kisaran 0 dan 1[21]. Persamaannya dapat dilihat pada persamaan (10).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (10)$$

b. *Standard Scaling*

Metode normalisasi *Zero-Mean* didasarkan pada *mean* dan standar deviasi. Standarisasi suatu *dataset* melibatkan pengubahan skala distribusi nilai, sehingga nilai rata-rata (*mean*) yang diamati adalah 0 dan standar deviasi adalah 1[21]. Persamaannya dapat dilihat pada persamaan (11).

$$x_{scaled} = \frac{x - x_{mean}}{x_{std}} \quad (11)$$

c. **Robust Scaling**

Scaler ini menghilangkan median dan menskalakan data sesuai dengan rentang kuartil antara kuartil ke-1 dan kuartil ke-3. Persamaannya dapat dilihat pada persamaan (12)

$$x_{scaled} = \frac{x - Q_1(x)}{Q_3(x) - Q_1(x)} \quad (12)$$

3.2.4. K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan metode *supervised learning*. *Instance* baru diklasifikasikan berdasarkan sampel *training* terdekat di ruang fitur.[6] KNN adalah metode klasifikasi data untuk memperkirakan kemungkinan suatu titik data akan menjadi anggota dari suatu kelompok berdasarkan pada kelompok yang terdekat dengan titik data tersebut.

KNN memiliki konsep dasar yang sangat simpel dan hanya terdiri dari dua Langkah, yaitu pelatihan dan klasifikasi. Pelatihan dalam KNN berfungsi untuk menyimpan setiap pola latih tanpa adanya proses pembangunan model pengklasifikasi seperti pada *Artificial Neural Network*. Proses pelatihan KNN menghasilkan k yang memberikan akurasi tertinggi dalam menggeneralisasi data-data yang akan datang. Namun, nilai k tidak dapat ditentukan secara matematika atau dengan kata lain proses pelatihan melakukan observasi terhadap sejumlah k sampai dihasilkan k yang paling optimum. Proses klasifikasi berfungsi untuk mengklasifikasi sebuah pola, KNN harus memeriksa semua pola latih untuk menemukan sejumlah k pola terdekat[22]. Adapun kelebihan dan kelemahan pada metode KNN adalah sebagai berikut:

- a. Kelebihan KNN
 KNN memiliki kelebihan yaitu tangguh terhadap *training* data yang *noise* dan efektif apabila data latihnya besar.
- b. Kelemahan KNN
 1. KNN perlu menentukan nilai dari parameter k (jumlah dari tetangga terdekat).
 2. Pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik.
 3. Biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap

sampel uji pada keseluruhan sampel *training*

Untuk melakukan klasifikasi, KNN menggunakan klasifikasi tetangga sebagai nilai prediksi dari sampel uji yang baru[23]. Peringkat untuk k tetangga terdekat dihitung menggunakan berbagai metode di antaranya, *euclidean distance*, *Manhattan distance*, *minkowski distance*, dan *cosine similarity*. Perhitungan *Minkowski Distance* dapat dilihat pada Persamaan (13).

$$D_{Mink}(x, y) = \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^p} \quad (13)$$

Perhitungan *Euclidean Distance* dapat dilihat pada Persamaan (14).

$$ED(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \quad (14)$$

Perhitungan *Manhattan Distance* dapat dilihat pada Persamaan (15).

$$MD(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (15)$$

Perhitungan *Cosine distance* dapat dilihat pada Persamaan (16).

$$CosD(x, y) = 1 - \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (16)$$

Dengan $D(x,y)$ adalah jarak, X_i adalah sampel data, Y_i adalah data, n adalah dimensi data, i adalah variabel data dan p adalah nilai positif, jika $p=2$ maka jarak menjadi *Euclidean distance*, jika $p=1$ maka jarak menjadi *Manhattan distance*.

Adapun Algoritma KNN dapat dituliskan sebagai berikut[23]:

- a. Tentukan k tetangga terdekat
- b. Hitung jarak data baru (*input*) dengan data *training*
- c. Urutkan jarak dari yang terdekat
- d. Periksa kelas k tetangga terdekat
- e. Kelas data baru = kelas mayoritas tetangga terdekat.

3.2.5. Cross Validation

Cross validation adalah pengujian standar yang dilakukan untuk memprediksi *error rate*. *Data training* dibagi secara acak ke dalam beberapa bagian dengan perbandingan yang sama kemudian *error rate* dihitung bagian demi bagian, selanjutnya hitung rata-rata seluruh *error rate* untuk mendapatkan *error rate* secara keseluruhan[24].

3.2.6. Confusion Matrix

Confusion matrix didefinisikan sebagai matriks yang menyediakan campuran *instance* kelas yang diprediksi dengan kelas yang sebenarnya. *Confusion*

matrix digunakan untuk mengevaluasi kinerja algoritma terhadap permasalahan tertentu dan memungkinkan untuk mendefinisikan performa matriks yang lebih luas seperti *accuracy*, *precision*, dan *recall*[25]. Tabel *confusion matrix* dapat dilihat dalam tabel I dan II.

True Positives (TP) menunjukkan prediksi bernilai benar sesuai dengan kondisi yang sesungguhnya bernilai benar. *False Positives* (FP) menunjukkan prediksi bernilai benar di mana pada kondisi yang sesungguhnya bernilai salah. *True Negatives* (TN) menunjukkan prediksi bernilai salah sesuai dengan kondisi yang sesungguhnya bernilai salah. *False Negatives* (FN) menunjukkan prediksi bernilai salah di mana pada kondisi yang sesungguhnya bernilai benar.

Pada tabel 2.2 multiclass confusion matrix, kolom yang berwarna kuning akan mengindikasikan semua nilai True Positive karena menunjukkan Nilai sebenarnya yang bernilai benar dan nilai prediksi yang bernilai benar dan untuk baris yang berwarna hijau mengindikasikan semua barisnya bernilai False Negative, pada nilai sebenarnya akan selalu bernilai benar karena menunjukkan kelas dirinya sendiri namun pada nilai prediksi akan selalu bernilai salah karena menunjukkan kelas yang lain.

TABEL I. BINARY CONFUSION MATRIX

		Nilai Prediksi	
		Benar	Salah
Nilai Sebenarnya	Benar	<i>True Positive</i>	<i>False Negative</i>
	Salah	<i>False Positive</i>	<i>True Negative</i>

TABEL II. MULTICLASS CONFUSION MATRIX

		Nilai Prediksi			
		C ₁	C ₂	...	C _N
Nilai Sebenarnya	C ₁	True Positive	False Negative	False Negative	False Negative
	C ₂	False Negative	True Positive	False Negative	False Negative
	...	False Negative	False Negative	True Positive	False Negative
	C _N	False Negative	False Negative	False Negative	True Positive

Accuracy adalah nilai yang menunjukan tingkat kedekatan antara nilai prediksi sistem dengan nilai prediksi manusia[26]. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, *accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan

keseluruhan data. Dengan kata lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai *accuracy* untuk *multiclass confusion matrix* dapat diperoleh dengan persamaan (17).

$$Accuracy = \frac{\sum_{i=1}^N TP(C_i)}{\sum_{i=1}^N \sum_{j=1}^N C_{i,j}} \quad (17)$$

Precision adalah nilai sensitifitas atau nilai ketepatan sistem antara informasi yang diberikan oleh sistem untuk menunjukkan secara benar data kelas negatif atau positif[26]. *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Maka, *precision* merupakan rasio prediksi yang bernilai benar positif (*True Positive*) terhadap keseluruhan hasil diprediksi yang bernilai positif. Dari semua kelas positif yang telah diprediksi dengan benar, berapa banyak data yang sebenarnya yang benar positif. Nilai *precision* dapat diperoleh dengan persamaan (18).

$$Precision\ of\ Class\ C_i = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \quad (18)$$

Recall adalah nilai yang menunjukkan tingkat keberhasilan atau spesifisitas untuk mengetahui kembali sebuah informasi secara benar tentang data yang kelas negatif ataupun positif[26]. *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* merupakan rasio prediksi yang bernilai benar positif (*True Positive*) terhadap keseluruhan data sebenarnya yang bernilai positif. Nilai *recall* dapat diperoleh dengan persamaan (19)[2].

$$Recall\ of\ Class\ C_i = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (19)$$

3.3. Pengujian

Pengujian skenario dilakukan dengan beberapa kondisi, adapun kondisi dalam pengujian skenario antara lain sebagai berikut:

- Ekstraksi fitur MFCC, delta dan delta-delta dengan variasinya, masing-masing 13 koefisien
- Durasi setiap *file* 60s, Panjang *frame* 40ms, dan *overlap* 40%
- Scaling* data menggunakan *Standard scaling*, *Min-Max Scaling*, dan *Robust Scaling*.
- Metode jarak menggunakan *Minkowski distance*, *Euclidean distance*, *Manhattan distance*, dan *Cosine distance*.
- Nilai *k* ditentukan menggunakan *K-fold cross validation*.
- Membandingkan waktu komputasi dengan metode penelitian yang sudah dilakukan.

4. HASIL DAN PEMBAHASAN

4.1. Pengaruh fitur dan *scaling data*

Pengujian ini bertujuan untuk mengetahui pengaruh fitur dan *scaling data* untuk klasifikasi sinyal audio. Pada penelitian ini terdapat delta dan delta 2 dari MFCC yang digunakan sebagai upaya untuk meningkatkan akurasi. Sehingga pengujian dilakukan untuk memilih fitur yang tepat dan menghasilkan hasil model yang terbaik. Proses *scaling* juga dilakukan untuk mengetahui pengaruh terhadap data.

Dari tabel III dapat diketahui kombinasi fitur MFCC-Delta-Delta 2 mendominasi dengan menjadi fitur yang tertinggi pada dua metode *scaler* yang berbeda yaitu pada min-max scaler dan *standard scaler* masing-masing memiliki akurasi sebesar 75% dan 80.5%. Pada robust scaler, kombinasi fitur MFCC-Delta menjadi yang terbaik dengan akurasi 80%. Pada eksperimen ini akurasi yang tinggi diikuti dengan *precision* yang tinggi pula dan untuk *recall* pada hasil eksperimen ini memiliki nilai yang sama dengan akurasinya.

TABEL III. MULTICLASS CONFUSION MATRIX

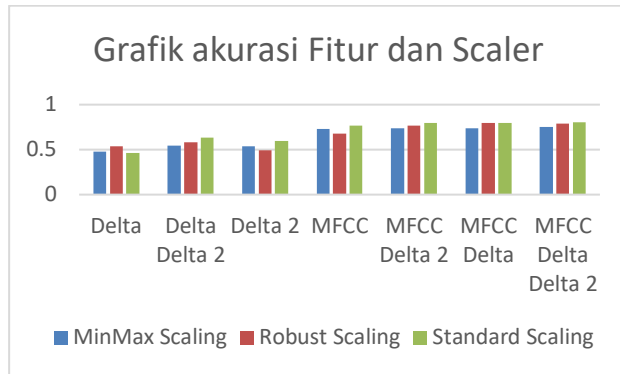
Fitur	Scaler	Akurasi (%)	Precisi-on (%)	Recall (%)
MFCC Delta Delta2	MinMax	75	78.51	75
MFCC Delta2	MinMax	74	76.98	74
MFCC Delta	MinMax	73.5	76.97	73.5
MFCC	MinMax	73	77.20	73
Delta Delta2	MinMax	54.5	58.80	54.5
Delta2	MinMax	53.5	52.89	53.5
Delta	MinMax	48	46.21	48
MFCC Delta	Robust	80	82.94	80
MFCC Delta Delta2	Robust	79	81.85	79
MFCC Delta2	Robust	77	80.11	77
MFCC	Robust	67.5	69.73	67.5
Delta Delta2	Robust	58.5	59.39	58.5
Delta	Robust	54	55.49	54
Delta2	Robust	49.5	47.67	49.5
MFCC Delta Delta2	Standard	80.5	83.21	80.5
MFCC Delta2	Standard	80	82.71	80
MFCC Delta	Standard	80	82.82	80
MFCC	Standard	77	80.15	77
Delta Delta2	Standard	63	63.57	63
Delta2	Standard	60	61.73	60
Delta	Standard	46	45.25	46

Akurasi untuk MFCC yang berdiri sendiri pada setiap metode *scaling* memiliki hasil yang lebih buruk dibandingkan dengan MFCC yang dikombinasikan dengan delta-deltanya. Begitu pula yang terjadi pada *precision* dan *recall*, delta MFCC juga meningkatkan performanya. Hal yang sama juga terjadi pada delta MFCC yang berdiri sendiri tanpa dikombinasikan MFCC hasilnya tidak cukup efektif sesuai dengan tabel 4.3. Hal ini membuktikan bahwa Delta MFCC dapat meningkatkan performa klasifikasi[8][11][29]. Karena Delta MFCC menunjukkan ciri sinyal dinamis sedangkan MFCC merepresentasikan ciri statis dan mempertimbangkan data penelitian ini berupa sinyal audio musik. Sehingga dengan mengombinasikan MFCC dengan delta-deltanya dapat meningkatkan performa akurasi dibandingkan tanpa delta-deltanya.

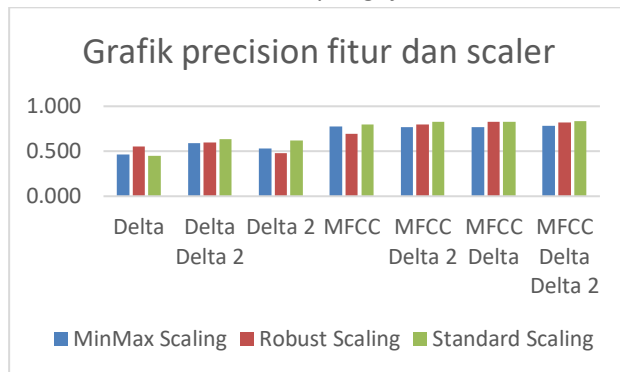
Untuk metode *scaling*, metode yang mendapatkan akurasi tertinggi pada fitur MFCC-Delta-Delta 2 adalah *standard scaling* sebesar 80,5% diikuti oleh *robust scaling* dan *min-max scaling* masing-masing sebesar 79% dan 75%. Sedangkan pada fitur MFCC-Delta akurasi tertinggi didapatkan oleh metode *standard scaling* dan *Robust scaling* dengan nilai sebesar 80%. Untuk *precision* pada MFCC-Delta-Delta 2 nilai tertinggi adalah dengan menggunakan *Standard scaling* selisih kurang dari 1% dari *robust scaling* sedangkan pada MFCC-Delta nilai tertingginya adalah dengan menggunakan *Robust scaling* dengan selisih juga kurang dari 1% dengan *Standard scaling*.

Adapun jika mengambil rata-rata akurasi, *precision* dan *recall* dari setiap fitur dengan metode *scaling* yang sama maka akurasi *Standard scaling* menjadi 69,5% sama dengan *recall*-nya dan *precision* menjadi 71,3%. Untuk *Robust Scaling* mendapatkan nilai sebesar 66,5% untuk akurasi dan *recall*, dan *precision* sebesar 68,2%. *Min-max scaling* mendapatkan akurasi dan *recall* yang sama sebesar 64,5% dan *precision* 66,8%.

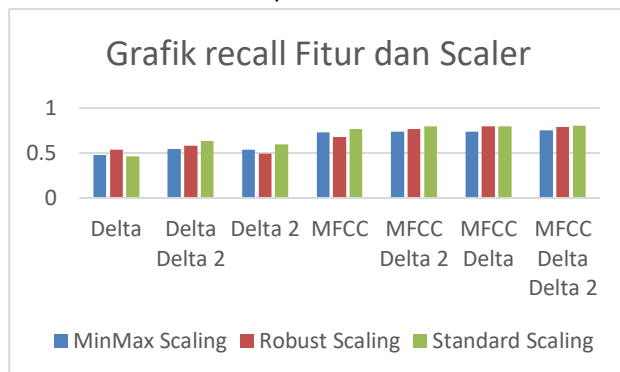
Perbedaan setiap *scaling* pada eksperimen ini tidak terlalu signifikan seperti terlihat pada gambar 6, 7 dan 8. Namun demikian menurut Ahsan[30], tidak terdapat *scaling* yang spesifik untuk semua *dataset*, maka dari itu peneliti butuh untuk menemukan *scaling* yang tepat dengan melakukan eksperimen. Pada penelitian ini, performa *scaling data* yang memiliki hasil yang lebih tinggi adalah *standard scaling* dengan melihat rata-rata dari akurasi, *precision* dan *recall*.



Gambar 6. Grafik akurasi pengujian fitur dan scaler



Gambar 7. Grafik precision fitur dan scaler



Gambar 8. Grafik recall pengujian fitur dan scaler

4.2. Pengaruh Jarak

Pada penelitian yang menggunakan KNN sebagai pengklasifikasinya, KNN memiliki dua parameter untuk diuji, yaitu metode jarak dan nilai k . Pada tahap ini, pengujian yang dilakukan adalah mencari metode jarak terbaik. Pengujian ini bertujuan untuk mengetahui pengaruh metode jarak tetangga pada KNN untuk dapat menghasilkan hasil model yang terbaik.

Pada tabel IV mencari metode jarak dengan menggunakan empat parameter teratas dari pengujian sebelumnya. Dapat diketahui hasil uji yang memiliki akurasi tertinggi adalah hasil uji yang menggunakan metode *manhattan* dan fitur MFCC-Delta dan MFCC-Delta-Delta2 dengan akurasi 85,5%.

Untuk akurasi tertinggi pada metode *euclidean* fitur MFCC-Delta-Delta2 sebesar 80,5% dan MFCC-Delta-Delta2 mendapatkan hasil akurasi tertinggi pada *cosine* sebesar 79,5%.

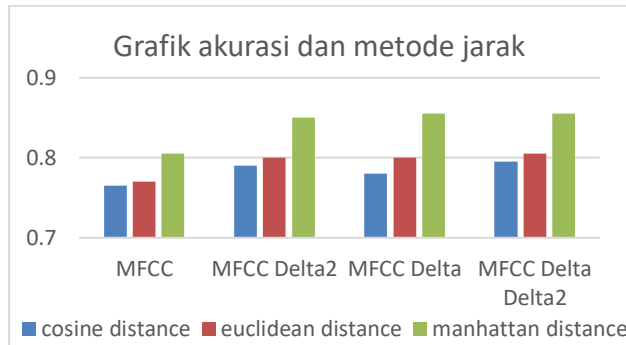
TABEL IV. HASIL PENGUJIAN JARAK

Fitur	Distance	Akurasi (%)	Precision (%)	Recall (%)
MFCC Delta Delta2	<i>cosine</i>	79.5	82.52	79.5
MFCC Delta2	<i>cosine</i>	79	82.27	79
MFCC Delta	<i>cosine</i>	78	81.5	78
MFCC	<i>cosine</i>	76.5	80.8	76.5
MFCC Delta Delta2	<i>euclidean</i>	80.5	83.21	80.5
MFCC Delta2	<i>euclidean</i>	80	82.71	80
MFCC Delta	<i>euclidean</i>	80	82.82	80
MFCC	<i>euclidean</i>	77	80.15	77
MFCC Delta Delta2	<i>manhattan</i>	85.5	87.14	85.5
MFCC Delta	<i>manhattan</i>	85.5	87.34	85.5
MFCC Delta2	<i>manhattan</i>	85	86.84	85
MFCC	<i>manhattan</i>	80.5	82.81	80.5

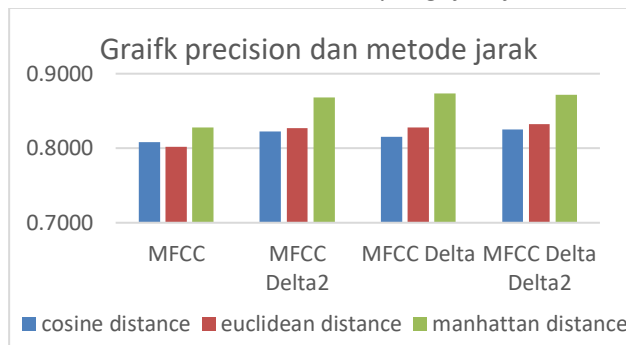
Untuk *precision* pada metode *manhattan* nilai tertinggi 87,34% dihasilkan oleh fitur MFCC-Delta. Pada metode *euclidean* nilai tertinggi pada 83,21% dihasilkan MFCC-Delta-Delta2. Pada metode *cosine* nilai tertinggi 82,52% dihasilkan fitur MFCC-Delta-Delta2. Sedangkan untuk *recall*, nilainya sama dengan akurasinya sesuai pada tabel IV.

Metode *manhattan* mendapatkan nilai akurasi yang signifikan dibandingkan dengan dua metode lainnya pada fitur yang sama seperti pada gambar 9. Metode *manhattan* memiliki akurasi dengan jarak 3-5% jika dibandingkan dengan metode *Euclidean* dan *cosine* pada fitur yang sama. Hal yang sama juga terdapat pada gambar 10 dan 11. Metode *manhattan* terlihat mendapat nilai yang signifikan dibandingkan dengan metode lainnya.

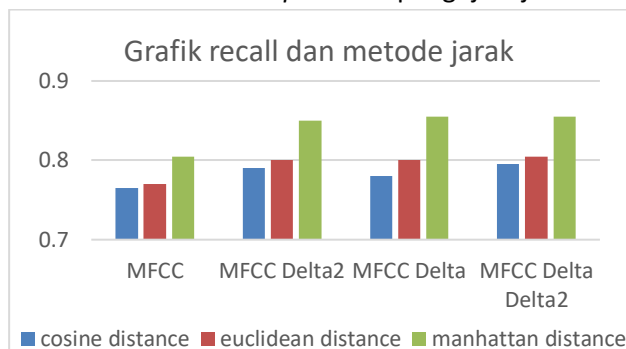
Konsep perhitungan jarak *euclidean* yang menerapkan konsep *pythagoras*, dengan mencari jarak terdekat secara geometri. Sedangkan, *manhattan* menerapkan konsep pencarian selisih murni antar data[31]. Walaupun nilai jarak pada *Euclidean* lebih kecil dibandingkan dengan *manhattan* namun pada penelitian ini *manhattan* lebih sesuai dengan data pada penelitian ini. Dimensi data juga berpengaruh terhadap performa metode jarak. *manhattan* memberikan kontras relatif yang jauh lebih tinggi daripada *Euclidean*[32].



Gambar 9. Grafik akurasi pengujian jarak



Gambar 10. Grafik precision pengujian jarak



Gambar 11. Grafik recall pengujian jarak

4.3. Nilai optimal K dan waktu komputasi

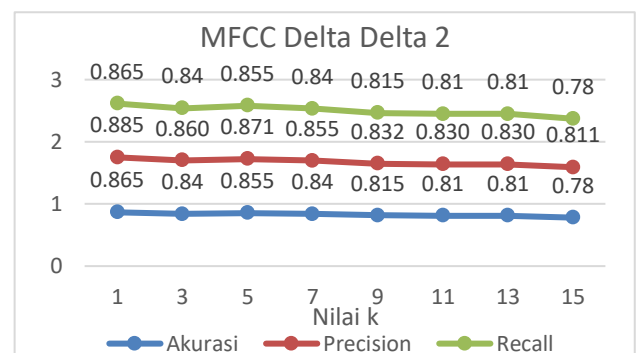
Pada pengujian ini dilakukan untuk mencari nilai k yang paling optimal. Nilai k dicari menggunakan *10-fold cross validation*, dataset dibagi menjadi sepuluh bagian. Masing-masing bagian akan bergantian menjadi *training data* dan *testing data*. Satu bagian menjadi *testing data* dan Sembilan lainnya menjadi *training data*. Hal ini dilakukan sampai sepuluh kali, sehingga setiap bagian akan mendapat kesempatan untuk menjadi *testing data*. Kemudian akan diambil rata-ratanya. Pada tahap ini, dihitung juga waktu komputasi yang dibutuhkan untuk mendapatkan hasil perhitungan

Pada tabel 4.5 didapatkan nilai k terbaik untuk masing-masing fitur. Fitur MFCC-Delta-Delta 2, MFCC-Delta dan MFCC-Delta 2 ketiganya memiliki nilai k optimal yang sama yaitu k=5. Untuk akurasi dan recall

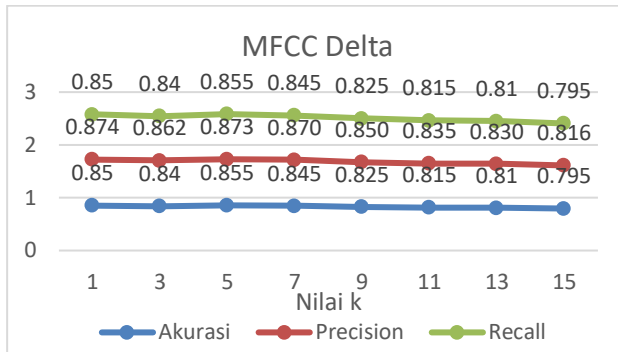
MFCC-Delta-Delta 2 dan MFCC-Delta memiliki besar yang sama yaitu sebesar 85.5% namun dengan jumlah fitur yang berbeda yaitu masing-masing 39 dan 26 fitur. Untuk *precision*, MFCC-Delta memiliki performa yang lebih baik yaitu 87.34% dibandingkan dengan MFCC-Delta-Delta 2 sebesar 87.14%. Untuk fitur MFCC-Delta2, akurasi dan *recall* mendapat nilai 85% sedangkan *precision* 86,64%. Untuk fitur MFCC, akurasi tertinggi mendapat nilai 81,5% pada k=6, *precision* 84,01% dan *recall* 81,5%.

TABEL V. HASIL PENGUJIAN NILAI K DAN WAKTU KOMPUTASI

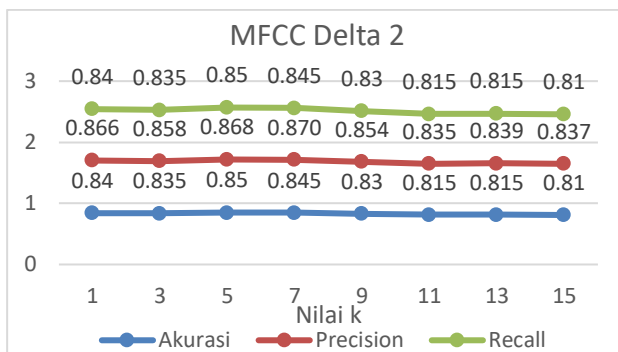
Fitur	MFCC Delta Delta2	MFCC Delta	MFCC Delta2	MFCC
Scaler	Standard	Standard	Standard	Standard
Jumlah Fitur	39	26	26	13
Distance	manhattan	manhattan	manhattan	manhattan
Nilai k	5	5	5	6
Accuracy (%)	85.5	85.5	85	81.5
Precision (%)	87.14	87.34	86.84	84.01
Recall (%)	85.5	85.5	85	81.5
fit time (s)	0.00399	0	0.00598	0.00698
score time (s)	0.02597	0.02797	0.02496	0.02398



Gambar 12. Grafik akurasi, precision dan recall terhadap nilai k MFCC Delta Delta 2



Gambar 13. Grafik akurasi, precision dan recall terhadap nilai k MFCC Delta



Gambar 14. Grafik akurasi, precision dan recall terhadap nilai k MFCC Delta 2

Pada Gambar 4.6, 4.7 dan 4.8 merupakan grafik sebaran akurasi, *precision* dan *recall* terhadap nilai k. Pada nilai k=1 ketiga fitur memiliki nilai akurasi yang lebih tinggi dibandingkan nilai k yang lain, fitur MFCC-Delta-Delta 2 memiliki akurasi sebesar 86,5% dengan nilai *precision* 88,5% dan *recall* 86,5%, MFCC-Delta akurasi 85%, *precision* 87,4% dan *recall* 85% dan MFCC-Delta 2 memiliki akurasi 84%, *precision* 86,6% dan *recall* 84%. Namun demikian, dengan nilai k=1 menyebabkan data akan rentan terhadap adanya *noise*. Maka akan dilihat nilai k selanjutnya. Didapatkan nilai k=5 yang merupakan nilai tertinggi pada ketiga fitur setelah k=1. Pada nilai k=5 *noise* akan lebih sedikit dibandingkan dengan k=1, maka nilai k yang optimal pada penelitian ini adalah k=5.

Hasil penelitian ini dibandingkan juga dengan metode klasifikasi lain yaitu BPNN yang pernah dilakukan oleh penelitian[2] serupa mengenai klasifikasi *mood* musik dan dengan *dataset* yang sama. Hasilnya didapatkan akurasi sebesar 87,6% lebih tinggi dibandingkan KNN yang menghasilkan 85.5%. Namun demikian BPNN tersebut membutuhkan waktu komputasi yang jauh lebih lama yaitu 901 detik. Sedangkan, KNN hanya membutuhkan waktu 0.02 detik. Dengan waktu yang jauh lebih

singkat KNN mampu mendapatkan akurasi yang mendekati penelitian sebelumnya yang menggunakan BPNN.

Hal ini dapat terjadi karena berbagai hal, di antaranya pendekatan dalam ekstraksi fitur, walaupun pada penelitian tersebut juga menggunakan MFCC, namun pemrosesan fiturnya berbeda, pada penelitian sebelumnya fitur yang digunakan mencapai 50.020 fitur untuk setiap sinyal audio sedangkan pada penelitian ini hanya menggunakan 13-39 fitur saja, sehingga membuat waktu komputasinya menjadi sangat jauh berbeda terutama pada bagian *training*. Neural Network membutuhkan waktu training yang cukup lama sedangkan KNN tidak melakukan *training* data sebelumnya karena bersifat *lazy learner*.

TABEL VI. PERBANDINGAN AKURASI DAN KOMPUTASI BERBAGAI METODE

Metode	Jumlah Fitur	Akurasi (%)	Waktu komputasi (s)
BPNN (penelitian lain)	50020	87.6	901.736
KNN (MFCC+Delta)	26	85.5	0.028
KNN (MFCC+Delta+Delta 2)	39	85.5	0.030
BPNN (MFCC+Delta)	26	87.5	132.343
BPNN (MFCC+Delta+Delta 2)	39	89.5	121.263

Selanjutnya eksperimen dilakukan menggunakan BPNN dengan jumlah fitur yang sama seperti pada penelitian ini untuk membandingkan hasilnya dengan KNN. Hasilnya seperti pada tabel 4.6, pada BPNN dengan MFCC-Delta yaitu sebanyak 26 fitur, akurasinya mendapatkan nilai 87,5%. Selain itu, BPNN dengan MFCC-Delta-Delta2 berjumlah 39 fitur mendapat akurasi yang lebih tinggi lagi yaitu sebesar 89,5%. Hasil tersebut menyamai bahkan lebih baik dibandingkan BPNN dengan 50.020 fitur dan dengan waktu komputasi yang lebih cepat. Hal ini juga dapat membuktikan penggunaan delta MFCC berpengaruh terhadap performa klasifikasi karena BPNN pada penelitian sebelumnya tersebut hanya menggunakan MFCC.

Hasil akurasi BPNN pada eksperimen ini meningkat dan waktu komputasi juga menurun. Namun jika dibandingkan dengan KNN, BPNN masih membutuhkan waktu komputasi yang lebih lama akibat adanya proses training data dan BPNN membutuhkan memori yang lebih besar dibandingkan KNN, namun BPNN memiliki akurasi yang lebih baik.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa pada metode KNN, jarak berpengaruh terhadap hasil klasifikasi, metode *manhattan* mendapatkan hasil yang lebih baik dibandingkan dengan *Euclidean* dan *cosine*. Dari hasil penelitian dapat diperoleh akurasi hasil *cross validation* KNN sebesar 85,5%, *precision* 87,34% dan *recall* 85,5% dengan nilai $k=5$ dan dengan metode jarak menggunakan *Manhattan*.

Pemilihan fitur dan *Scaling data* berpengaruh terhadap hasil klasifikasi, penggunaan delta MFCC pada klasifikasi menambah akurasi model yang membuat MFCC-Delta menjadi model yang terbaik. Sedangkan *standard scaling* memiliki akurasi yang lebih tinggi dibandingkan dengan *robust* dan *min-max scaling*.

Saran yang penulis berikan jika dilakukannya pengembangan atas penelitian ini yaitu memperbanyak jumlah data dan varian yang lebih beragam dan dengan variasi tipe musik yang memiliki vokal suara manusia. Untuk klasifikasi data dapat dibandingkan dengan metode lainnya.

UCAPAN TERIMA KASIH

Terima kasih saya ucapkan kepada seluruh rekan-rekan saya yang telah membantu dan berkontribusi dalam penelitian saya ini.

DAFTAR PUSTAKA

- [1] Q. Meng, J. Jiang, F. Liu, and X. Xu, "Effects of the musical sound environment on communicating emotion," *Int. J. Environ. Res. Public Health*, vol. 17, no. 7, pp. 1–19, 2020, doi: 10.3390/ijerph17072499.
- [2] P. I. Maulana, A. Aranta, and F. Bimantoro, "Klasifikasi Mood Musik berdasarkan Mel Frequency Cepstral Coefficients dengan Backpropagation Neural Network," 2021.
- [3] A. Shakya, B. Gurung, M. S. Thapa, and M. Rai, "Music Classification Based on Genre and Mood," pp. 168–183, 2017, doi: 10.1007/978-981-10-6430-2.
- [4] D. Ayu and P. Bayu, "Hierarchical SVM-kNN to Classify Music Emotion," pp. 5–10, 2020.
- [5] R. Sarno, J. A. Ridoean, D. Sunaryono, and Wijaya, "Classification of Music Mood Using MPEG-7 Audio Features and SVM with Confidence Interval," *Int. J. Artif. Intell. Tools*, 2018, doi: 10.1142/S0218213018500161.
- [6] R. Thiruvengatanadhan, "Speech / Music Classification using MFCC and KNN," *Int. J. Comput. Intell. Res.*, vol. 13, no. 10, pp. 2449–2452, 2017.
- [7] D. Anggraeni, W. S. M. Sanjaya, M. Y. S. Nurasyidiek, and M. Munawwaroh, "The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) method based on Python to Control Robot Arm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, 2018, doi: 10.1088/1757-899X/288/1/012042.
- [8] M. A. Hossan, S. Memon, and M. A. Gregory, "A novel approach for MFCC feature extraction," *4th Int. Conf. Signal Process. Commun. Syst. ICSPCS'2010 - Proc.*, 2010, doi: 10.1109/ICSPCS.2010.5709752.
- [9] I. G. Harsemadi and I. M. Sudarma, "Penggolongan Musik Terhadap Suasana Hati Menggunakan Metode K-Means," in *Konferensi Nasional Sistem & Informatika 2017*, 2017, pp. 49–54.
- [10] S. A. A. Yusuf and R. Hidayat, "MFCC feature extraction and KNN classification in ECG signals," *2019 6th Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2019*, pp. 1–5, 2019, doi: 10.1109/ICITACEE.2019.8904285.
- [11] Q. Jiang, F. Chang, and B. Sheng, "Bearing Fault Classification Based on Convolutional Neural Network in Noise Environment," *IEEE Access*, vol. 7, pp. 69795–69807, 2019, doi: 10.1109/ACCESS.2019.2919126.
- [12] 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," in *Seminar Nasional Teknologi Informasi dan Komunikasi Terapan (SEMANTIK) 2015*, 2015, pp. 239–243.
- [13] M. S. W. S and Z. Salleh, "Implementasi Pengenalan Pola Suara menggunakan Mel-Frequency Cepstrum Coefficients (Mfcc) dan Adaptive Neuro-Fuzzy Inferense System (Anfis) sebagai Kontrol Lampu Otomatis," *Al-HAZEN J. Phys.*, vol. 1, no. 1, pp. 43–54, 2014.
- [14] L. Halimah, "Musik Dalam Pembelajaran," *EduHumaniora J. Pendidik. Dasar Kampus Cibiru*, vol. 2, no. 2, pp. 1–9, Jul. 2016, doi: 10.17509/eh.v2i2.2763.
- [15] W. Chijioke, "Predicting Listener'S Mood Based on Music Genre: an Adapted Reproduced Model of Russell and Thayer," *J. Technol. Manag. Bus.*, vol. 4, no. 1, pp. 39–58, 2017.
- [16] I. G. Harsemadi, "Implementasi Fast Fourier Transform Pada Ekstraksi Fitur Mood Dalam Musik," *Proc. Semin. Nas. Multidisiplin Ilmu Univ. Budi Luhur*, no. April, pp. 121–129, 2017.
- [17] U. E. Akpudo and J. W. Hur, "A cost-efficient mfcc-based fault detection and isolation technology for electromagnetic pumps," *Electron.*, vol. 10, no. 4,

- pp. 1–21, 2021, doi: 10.3390/electronics10040439.
- [18] G. Kour and N. Mehan, "Music Genre Classification using MFCC, SVM and BPNN," *Int. J. Comput. Appl.*, vol. 112, no. 6, pp. 12–14, 2015.
- [19] H. D. Arani, "Identifikasi Ayat pada Bacaan Menggunakan Metode Dynamic Time Warping Berdasarkan Fitur Mel Frequency Cepstral Coefficient untuk Sistem Tutorial Hafalan Al-Quran," 2017.
- [20] A. H. Abriyono, "Pengenalan Ucapan Suku Kata Bahasa Lisan Menggunakan Ciri LPC, MFCC, dan JST," *IJCCS*, vol. 6, no. 2, pp. 23–34, 2013, doi: 10.22146/ijccs.2149.
- [21] A. Ambarwari, Q. J. Adrian, U. T. Indonesia, and Y. Herdiyeni, "Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman," *RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 1, pp. 117–122, 2020, doi: 10.29207/resti.v4i1.1517.
- [22] Suyanto, *Machine Learning Tingkat Dasar dan Lanjut*. Bandung: Informatika, 2018.
- [23] Santoso and M. I. Irawan, "Perbandingan Metode K-Nearest Neighbor (K-Nn) dan Learning Vector Quantization (Lvq) untuk permasalahan Classification Problem," 2016.
- [24] H. Leidiyana, "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor," *J. Penelit. Ilmu Komputer, Syst. Embed. Log.*, vol. 1, no. 1, pp. 65–76, 2013.
- [25] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies*, vol. 9, no. 4, p. 81, 2021, doi: 10.3390/technologies9040081.
- [26] M. Azhari, Z. Situmorang, and R. Rosnelly, "Perbandingan Akurasi, Recall, dan Presisi Klasifikasi pada Algoritma C4.5, Random Forest, SVM dan Naive Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 640, 2021, doi: 10.30865/mib.v5i2.2937.
- [27] V. L. Nguyen, D. Kim, V. P. Ho, and Y. Lim, "A New Recognition Method for Visualizing Music Emotion," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 3, pp. 1246–1254, 2017, doi: 10.11591/ijece.v7i3.pp1246-1254.
- [28] D. Griffiths, S. Cunningham, J. Weinel, and R. Picking, "A multi-genre model for music emotion recognition using linear regressors," *J. New Music Res.* ISSN, 2021, doi: 10.1080/09298215.2021.1977336.