IMPLEMENTASI METODE MFCC (MEL FREQUENCY CEPSTRAL COEEFICIENT) DAN NAIVE BAYESIAN UNTUK KLASIFIKASI NADA DASAR GITAR

Rizal Solehudin[1], Galih Hermawan, S.Kom., M.T.[2]

Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : rizalsolehudin92@yahoo.com[1], galih.hermawan@yahoo.co.id[2]

ABSTRAK

Gitar merupakan alat musik modern yang populer, dan banyak digunakan sebagai instrumen pengisi *rhythm* dan melodi dalam sebuah lagu. Pengetahuan kunci gitar atau akor diperlukan untuk memainkan gitar. Akor merupakan rangkaian nada yang membangun keharmonisasian pada musik [9]. Mengenali sebuah akor gitar lebih sulit dilakukan oleh indera pendengaran manusia khususnya bagi mereka yang awam terhadap akor gitar. Penelitian ini menggunakan metode MFCC (*Mel Frequency Cepstral Coefficient*) untuk ekstraksi suara dan Naive Bayesian untuk klasifikasi. Metode MFCC merupakan teknik ekstraksi ciri yang telah luas dipakai pada pemrosesan sinyal suara, terutama pada pengenalan pembicara. Metode MFCC mempunyai ciri-ciri khusus yang akan diekstrak dari masukan sinyal suara yang nantinya akan dikenali.

Berdasarkan hasil penelitian yang telah dilakukan dapat diketahui klasifikasi nada dasar gitar dengan klasifikasi Naïve Bayesian menggunakan metode MFCC sebagai metode untuk ekstraksi pengolah suara dan metode naive bayesian sebagai metode untuk klasifikasi. Dari hasil pengujian yang telah dilakukan klasifikasi nada dasar gitar menggunakan metode MFCC dan naive bayesian didapat tingkat akurasi sebesar 70%.

Kata Kunci: Nada Dasar Gitar, *Mel Frequency Cepstral Coefficient*, Naive Bayesian, Kecerdasan Buatan.

1. PENDAHULUAN

Gitar merupakan alat musik modern yang populer, dan banyak digunakan sebagai instrumen pengisi *rhythm* dan melodi dalam sebuah lagu. Pengetahuan kunci gitar atau akor diperlukan untuk memainkan gitar. Akor merupakan rangkaian nada yang membangun keharmonisasian pada musik [9]. Mengenali sebuah akor gitar lebih sulit dilakukan oleh indera pendengaran manusia khususnya bagi mereka yang awam terhadap akor gitar.

Salah satu solusi yang dapat dilakukan untuk menanggulangi masalah tersebut adalah dengan cara komputerisasi, yaitu penggunaan komputer sebagai sistem pengenalan akor pada alat gitar. Karakter komputer yang konsisten pada setiap eksekusi perintah yang diberikan, memudahkan pengguna sistem untuk mendapatkan hasil pengenalan akor gitar yang akurat.

Sebelumnya dilakukan penelitian oleh Tito Galit Permana, untuk identifikasi akor gitar menggunakan Algoritma Harmonic Product Spectrum di dapatkan tingkat akurasi sebesar 75,68% [1]. Pada penelitian yang dilakukan oleh Ardi Priagung, klasisfikasi genre pada lagu dengan metode Naive Bayes didapatkan tingkat akurasi sebesar 83,3%. [2]. Metode Naive Bayesian merupakan sebuah metode pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan.

Berdasarkan hal tersebut diperlukan penelitian lebih lanjut tentang mengklasifikasi akor gitar untuk meningkatkan akurasi dengan menggunakan metode yang berbeda. Penelitian ini menggunakan metode MFCC (*Mel Frequency Cepstral Coefficient*) untuk ekstraksi suara dan Naive Bayesian untuk klasifikasi. Metode MFCC merupakan teknik ekstraksi ciri yang telah luas dipakai pada pemrosesan sinyal suara, terutama pada pengenalan pembicara. Metode MFCC mempunyai ciri-ciri khusus yang akan diekstrak dari masukan sinyal suara yang nantinya akan dikenali. *Mel Frequency Ceptral Coefficient* (MFCC) adalah salah satu ekstraksi fitur yang paling berhasil dalam pengenalan suara. Pada penelitian yang dilakukan oleh Eko Riyanto dan Sutejo, perbandingan metode ekstraksi ciri suara MFCC, PS-ZCPA, DAN LPC hasil yang didapatkan adalah ekstraksi ciri MFCC mempunyai tingkat keakuratan yang lebih baik dan waktu ekstraksi ciri yang tercepat diantara ekstraksi ciri PS-ZCPA dan LPC [3].

Dari permasalahan dan solusi yang telah dijelaskan, maka penelitian ini akan mengklasifikasi akor gitar berdasarkan suara dengan menerapkan metode MFCC untuk proses ekstraksi suara dan metode *Naive Bayesian* untuk proses klasifikasi suara, diharapkan metode MFCC dan *Naïve Bayesian* dapat mengklasifikasi akor gitar

berdasarkan ekstraksi suara dan mengukur tingkat keakuratan klasifikasinya.

1.1 Nada Dasar Gitar

Chord atau akor secara umum dapat diartikan sebagai suatu rangkaian nada-nada yang tersusun secara teratur dari sebuah tangga nada dan bias mewakili tangga nada tersebut [11].Jika diartikan dalam gitar, akor adalah dua atau lebih nada yang dibunyikan bersamaan dalam satu petikan dan dalam satu waktu yang sama. Dalam sebuah lagu, akor berfungsi sebagai musik pengiring atau *rhythm*.

Secara umum, akor dibagi kedalam 3 kelompok yaitu : akor mayor, akor minor dan akor substitusi [10].

a. Akor Mayor

Akor mayor adalah beberapa nada diatonic mayor yang dibunyikan secara bersamaan, dalam satu waktu yang sama. Nada diatonis mayor adalah nada-nada dengan urutan jarak antar nada 1-1-1/2-1-1-1-1/2.

b. Akor Minor

Akor minor adalah beberapa nada diatonic minor yang dibunyikan secara bersamaan, dalam satu waktu yang sama. Nada diatonis mimor adalah nada-nada dengan urutan jarak antar nada 1-1/2-1-1-1/2-1-1.

c. Akor Substitusi

Akor substitusi tidak berasal dari tangga nada diatonis mayor maupun minor, tetapi lebih ke interval atau jarak nada yang menyusunnya.

Akor dasar gitar dapat dilihat pada gambar berikut:

Chord dasar	Mayor	Minor
C	CM	Cm
C#	C#M	C#m
D	DM	Dm
D#	D#M	D#m
E	EM	Em
F	FM	Fm
F#	F#M	F#m
G	GM	Gm
G#	G#M	G#m
A	AM	Am
A#	A#M	A#m
B	BM	Bm

Gambar 1 Akor Dasar Gitar

1.2 Kecerdasan Buatan

Kecerdasan buatan (Artificial inteligence) adalah salah satu cabang ilmu pengetahuan yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal ini biasanya dilakukan dengan mengikuti karakteristik dan analogi berpikir dari kecerdasan manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer.

Semakin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Karakteristik cerdas sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Kecerdasan Buatan tidak hanya merambah di berbagai disiplin ilmu yang lain. Irisan antara psikologi dan kecerdasan buatan melahirkan sebuah area yang dikenal dengan nama cognition & psycolinguistics. Irisan antara teknik elektro dengan kecerdasan buatan melahirkan berbagai ilmu seperti pengolahan citra, teori kendali, pengenalan pola dan robotika.

Kecerdasan buatan digunakan untuk menganalisis pemandangan dalam citra dengan perhitungan simbol-simbol yang mewakili isi pemandangan tersebut setelah citra diolah untuk memperoleh ciri khas. Kecerdasan buatan bisa dilihat sebagai tiga kesatuan yang terpadu yaitu persepsi, pengertian dan aksi. Persepsi menerjemahkan sinyal dari dunia nyata dalam citra menjadi simbol-simbol yang lebih sederhana, pengertian memanipulasi simbol-simbol tersebut untuk memudahkan penggalian suatu informasi tertentu, dan aksi menerjemahkan simbol-simbol yang telah dimanipulasi menjadi sinyal lain yang dapat merupakan hasil akhir.

1.3 Pengolahan Suara

Pengenalan suara merupakan salah satu upaya agar suara dapat dikenali atau diidentifikasi sehingga dapat dimanfaatkan. Pengenalan suara dapat dibedakan ke dalam tiga bentuk pendekatan, yaitu pendekatan akustik-fonetik (*the acoustic-phonetic approach*), pendekatan kecerdasan buatan (*the artificial intelligence approach*), dan pendekatan pengenalan-pola (*the pattern recognition approach*).

Berikut ini merupakan penjelasan dari masing-masing blok:

1. Pengekstraksi ciri.

Merupakan proses mendapatkan sederetan besaran pada bagian sinyal masukan untuk menetapkan pola pembelajaran atau pola uji. Pada sinyal suara, ciri-ciri besaran biasanya merupakan keluaran dari beberapa bentuk teknik analisis spektrum seperti *filter bank analyzer*, LPC atau DFT (*Discrete Fourier Transform*).

2. Pembelajaran Pola

Satu atau lebih pola pembelajaran yang berhubungan dengan bunyi suara dari kelas yang sama, digunakan untuk membuat pola representatif dari ciri-ciri kelas tersebut. Hasilnya yang biasa disebut dengan pola referensi, dapat menjadi sebuah model yang mempunyai karakteristik bentuk statistik dari ciri-ciri pola referensi.

3. Perbandingan dengan Pola Model

Pola uji yang akan dikenali, dibandingkan dengan setiap kelas pola referensi. Kesamaan besaran antara pola uji dengan setiap pola referensi akan dihitung.

4. Logic Decision

Menentukan kelas pola referensi mana yang paling cocok untuk pola uji berdasarkan klasifikasi pola.
Pengenalan suara secara umum dapat dibagi menjadi tiga tahap, yaitu : ekstraksi ciri, pemodelan, dan pengenalan. Ekstraksi ciri adalah upaya untuk memperoleh ciri dari sinyal suara.

1.4 Preprocessing

Preprocessing adalah proses untuk memudahkan dalam menghasilkan nilai dari ekstraksi ciri. Pada tahap ini Sinyal suara yang akan diproses bersifat analog sehingga jika akan dilakukan pengolahan secara digital, sinyal suara tersebut harus dikonversi menjadi sinyal digital, berupa urutan angka dengan tingkat presisi tertentu yang dinamakan analog to digital conversion dengan menggunakan analog-to-digital converter (ADC).

1.5 Akuisisi Data

Suara yang akan digunakan pada penelitian ini berasal dari suara 24 akor dasar gitar yang dimainkan dengan cara dipetik serentak dari atas ke bawah, kemudian direkam. Suara transisi 2 akor dari 24 akor akan diulang sebanyak 15 kali perekaman Untuk mendapatkan hasil perekaman yang terdapat sedikit *noise*, perekaman dilakukan pada ruangan yang hening. Pengambilan data suara dilakukan dengan cara merekam suara 2 akor gitar, setiap satu akor ke akor lainnya sebanyak 15 kali, sehingga masing-masing akor memiliki 10 data latih dan 5 data uji.

1.6 Metode MFCC (Mel Frequency Cepstral Coefficient))

MFCC (*Mel Frequency Cepstral Coefficients*) merupakan salah satu metode yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition*. Metode ini digunakan untuk melakukan *feature extraction*, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter [2].
.

Ada pun proses pengolahan sinyal pada ekstraksi ciri menggunakan MFCC:

1. Pre-emphasis

Pre-emphasis merupakan salah satu jenis filter yang sering digunakan sebelum sebuah signal diproses lebih lanjut. Filter ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara. Tujuan dari preemphasis filter ini adalah:

1. Mengurangi *noise ratio* pada sinyal, sehingga dapat meningkatkan kulitas sinyal.
2. Untuk mendapatkan bentuk spektral frekuensi sinyal wicara yang lebih halus.
3. Menyeimbangkan spektrum dari sinyal suara.

Pada saat memproduksi sinyal suara, glotis manusia menghasilkan sekitar -12 dB octave slope. Namun ketika energi akustik tersebut dikeluarkan melalui bibir, terjadi peningkatan sebesar +6 dB. Sehingga sinyal yang terekam oleh *microphone* adalah sekitar -6 dB *octave slope*. Dimana bentuk spektral yang relatif bernilai tinggi untuk daerah rendah dan cenderung turun secara tajam untuk daerah fekuensi diatas 2000 Hz. *Filter preemphasis* didasari oleh hubungan *input/output* dalam domain waktu yang dinyatakan dalam persamaan seperti berikut:

$$y(n) = s(n) - \alpha s(n-1) \quad (1)$$

dimana:
y(n) = Sinyal hasil *preemphasis*
s(n) = Sinyal sebelum *preemphasis*
α merupakan konstanta *filter preemphasis*, biasanya bernilai 0.97.

2. Frame blocking

Frame Blocking adalah pembagian suara menjadi beberapa frame yang nantinya dapat memudahkan dalam perhitungan dan analisa suara, satu frame terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi samplingnya [7]. Panjang frame yang digunakan, sangat mempengaruhi keberhasilan dalam analisa spektral. Disatu sisi, ukuran dari frame harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik. Tetapi di sisi lain, ukuran frame juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik. Refresentasi fungsi *frame blocking* sebagai berikut:

$$x(n) = y(M+n) \quad (2)$$

dimana,
x(n) =sinyal sesudah di *frame blocking*
y =sinyal hasil *pre-emphasis*
M = *overlapping frame*
n= 1,2,3,…

3. Windowing

Suara yang di potong-potong menjadi beberapa frame membuat data suara menjadi discontinue, hal ini mengakibatkan kesalahan data proses *fourier transform.* Agar tidak terjadi kesalahan data pada proses fourier transform maka sampel suara yang telah dibagi menjadi beberapa *frame* perlu dijadikan suara kontinu dengan cara mengalikan tiap frame dengan *window* tertentu[3]. Jenis *window* yang dipakai pada proses ini adalah *window hamming*. Berikut ini adalah representasi fungsi *window* terhadap signal suara yang diinputkan.

$$(n) = x(n)w(n) \quad (3)$$

rumus *window hamming* sebagai berikut:

$$w[n] = 0.54 - 0.46*\cos(2*\pi*n/(N-1)) \quad (4)$$

dimana:
n = 1,2,3,…
π = 3.14
N = panjang frame

Fungsi window yang paling sering digunakan dalam aplikasi *voice recognition* adalah *Window Hamming*. Fungsi *window hamming* ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu noise yang dihasilkan pun tidak terlalu besar (kurang lebih 1.36 BINS).

4. *Fast Fourier Transform* (FFT)

Fast Fourier Transform merupakan DFT dengan algoritma yang lebih optimal, sehingga menghasilkan perhitungan yang lebih cepat. Dengan DFT, memerlukan waktu O(n2) untuk mengolah sampel data sebanyak nbuah. Hal ini tentunya akan memakan waktu lama bila sampel data makin banyak. Proses pada metode *fast Fourier Transform* (FFT) adalah mengkonversi setiap *frame* N sampel dari domain waktu menjadi domain frekuensi, transformasi fourier merupakan metode yang efisien untuk transformasi fourier diskrit yang banyak digunakan pada pemrosesan sinyal digital [5].

Metode FFT dapat dilakukan dalam domain waktu dan frekuensi yang disebut sebagai desimasi – dalam – waktu *(decimation-in-time)* dan desimasi-dalam frekuensi *(decimation-in-frequency).* Pada prinsipnya algoritma ini adalah memecah N-titik menjadi dua (N/2)–titik, kemudian di pecah lagi pada tiap (N/2)– titik menjadi dua (N/4) –titik, begitu seterusnya sampai hanya terdapat 1 titik.

5. *Discrete Cosine Transform* (DCT)

Tahap ini melakukan konversi hasil *mel-frequency* ke domain waktu dengan menggunakan *Discrete Cosine Transform* (DCT). Berikut adalah persamaan yang digunakan dalam DCT.

$$C_j = \sum_{i=0}^{M} X_i \cos\left(j(i-1)2\left(\frac{\pi}{M}\right)\right) \quad (5)$$

Keterangan :
Cj = nilai koefisien C ke-j
j = jumlah koefisien cepstral
i = jumlah wrapping
Xi = hasil mel-frequency pada frekuensi ke-i

1.7 Klasifikasi Naïve Bayesian

Naïve bayesian adalah suatu metode pengklasifikasian paling sederhana dengan menggunakan peluang yang ada, dimana diasumsikan bahwa setiap variable X bersifat bebas (independence) [4]. Karena asumsi variabel tidak saling terikat, maka didapatkan :

$$P(X \mid Y = y) = \prod_{i=1}^{q} P(X_i \mid Y = y) \quad (6)$$

Terdapat beberapa langkah dalam pengklasifikasian menggunakan metode naive bayesian, berikut adalah langkah - langkahnya :

Training :

1. Hitung rata-rata (*mean*) tiap fitur dalam dataset training dengan.
   $$\mu = \frac{\sum x_i}{n} \quad (7)$$
   Dimana:
   μ = mean MFCC dataset
   n = banyaknya data
   $\sum x_i$ = jumlah nilai data

2. Kemudian hitung nilai varian dari dataset training tersebut seperti pada.
   $$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2 \quad (8)$$
   Dimana:
   σ^2= varians
   μ= mean
   x_i = nilai data
   n = banyaknya data

Testing :

1. Hitung probabilitas (*Prior*) tiap kelas yang ada dengan cara menghitung jumlah data tiap kelas dibagi jumlah total data secara keseluruhan.
2. Selanjutnya menghitung densitas probabilitasnya. Fungsi densitas mengekspresikan probabilitas relatif. Data dengan mean μ dan standar deviasi σ, fungsi densitas probabilitasnya adalah :
   $$\varphi_{\mu\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (9)$$
   Dimana :
   x = data masukan
   π = 3,14
   σ = standar deviasi
   μ = mean
3. Setelah didapatkan nilai densitas probabilitasnya, selanjutnya menghitung posterior masing-masing kelas dengan menggunakan persamaan.
   $$posterior = \frac{prior\ x\ likehood}{evidence}$$
   Atau bisa ditulis
   $$Posterior = \frac{P(C_i)x\,P(X1 \dots\dots\dots\dots\dots\dots\dots Xn|C_i)}{P(X1 \dots\dots\dots\dots\dots\dots\dots Xn|C_i)}$$
4. Setelah didapat nilai posterior masing-masing kelas, maka kelas yang sesuai untuk data masukan adalah kelas yang memiliki nilai posterior terbesar.

1.8 Pengujian *Confusion Matrix*

Pengujian yang dilakukan pada metode klasifikasi terdapat pada bagian akurasi dari hasil klasifikasi. Akurasi sebuah klasifikasi berpengaruh terhadap performa dari suatu metode klasifikasi. Untuk melakukan pengujian akurasi dapat digunakan *confusion matrix* yaitu sebuah matriks dari prediksi yang akan dibandingkan dengan kelas yang asli dari data masukan. Setiap kolom dari matriks berkorespondensi kepada hasil klasifikasi

dan setiap baris pada masukan. Akurasi sebuah klasifikasi dimana i=j menerangkan akurasi dari klasifikasi pada setiap kelas [9]. Berikut contoh *Confusion Matrix* dapat dilihat pada tabel 2.

Tabel 2 *Confusion Matrix*

Kelas		Hasil Klasifikasi	
		0	1
Target	0	00	01
	1	10	11

Untuk menghitung akurasinya digunakan formula :

$$\frac{jumlah\ akurasi}{total\ data} X\ 100\% = Akurasi\%$$

2. ISI PENELITIAN

2.1 Analisis Masalah

Gitar merupakan alat musik modern yang populer, dan banyak digunakan sebagai instrumen pengisi *rhythm* dan melodi dalam sebuah lagu. Pengetahuan kunci gitar atau akor diperlukan untuk memainkan gitar. Akor merupakan rangkaian nada yang membangun keharmonisasian pada musik [9]. Mengenali sebuah akor gitar lebih sulit dilakukan oleh indera pendengaran manusia khususnya bagi mereka yang awam terhadap akor gitar.

Salah satu solusi yang dapat dilakukan untuk menanggulangi masalah tersebut adalah dengan cara komputerisasi, yaitu penggunaan komputer sebagai sistem pengenalan akor pada alat gitar. Karakter komputer yang konsisten pada setiap
eksekusi perintah yang diberikan, memudahkan pengguna sistem untuk mendapatkan hasil pengenalan akor gitar yang akurat.

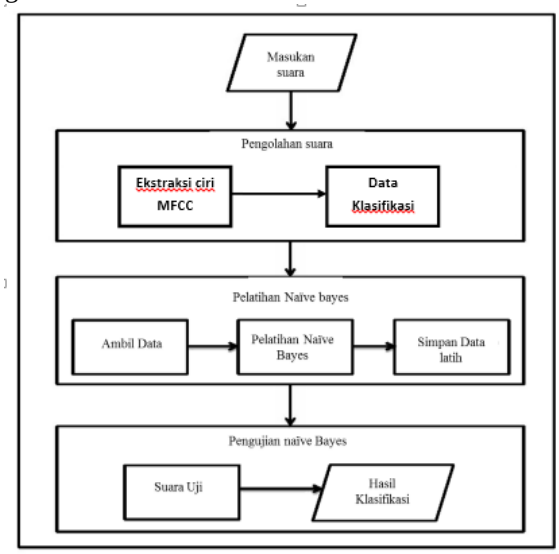
Sebelumnya dilakukan penelitian oleh Tito Galit Permana, untuk identifikasi akor gitar menggunakan Algoritma Harmonic Product Spectrum di dapatkan tingkat akurasi sebesar 75,68% [1]. Pada penelitian yang dilakukan oleh Ardi Priagung, klasisfikasi genre pada lagu dengan metode Naive Bayes didapatkan tingkat akurasi sebesar 83,3%. [2]. Metode Naive Bayesian merupakan sebuah metode pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan.

Berdasarkan hal tersebut diperlukan penelitian lebih lanjut tentang mengklasifikasi akor gitar untuk meningkatkan akurasi dengan menggunakan metode yang berbeda

Dari permasalahan dan solusi yang telah dijelaskan, maka penelitian ini akan mengklasifikasi akor gitar berdasarkan suara dengan menerapkan metode MFCC untuk proses ekstraksi suara dan metode *Naive Bayesian* untuk proses klasifikasi suara, diharapkan metode MFCC dan *Naïve Bayesian* dapat mengklasifikasi akor gitar berdasarkan ekstraksi suara dan mengukur tingkat keakuratan klasifikasinya.

2.2 Analisis Sistem

Analisis Sistem pada penelitian ini akan menjelaskan proses yang ada pada pada sistem klasifikasi nada dasar gitar dengan menggunakan gitar sebagai media input. Berikut adalah alur proses sistem yang akan dibangun dapat dilihat pada gambar 2 :



Gambar 2. Alur Analisis Sistem

2.3 DC Removal

Berikut adalah persamaan untuk menghitung DC Removal.

$$y(n) = x(n) - x, 0 \leq n \leq N-1$$

Contoh perhitungan *DC Removal* untuk akor C dengan sampel data sinyal di gambar berikut.

i	Signal ke i
0	-7
1	4
2	-3
3	2
4	8
5	12
6	14
7	11
8	4
9	-4
10	-6
11	-5
12	-9
13	-14
14	-15
15	6
16	10
17	-20
18	-24
19	-2
20	34
21	58
22	66
23	64

Hitung *DC Removal* :

(-7) +---+ (64)

=7.5

Setelah nilai rata-rata diketahui, kurangkan nilai Sinyal awal dengan hasil *DC Removal* sehingga nilai sinyal menjadi :

n0 = -7 – 7.5 = -14.5

hitung untuk data sinyal berikutnya dengan cara yang sama, setelah dilakukan proses *DC Removal* adalah

Tabel 2 Data *DC Removal*

i	Signal ke i
0	-14.5
1	-11.5
2	-10.5
3	-5.5
4	0.5
5	4.5
6	6.5
7	3.5
8	-3.5
9	3.5
10	-13.5
11	-12.5
12	-16.5
13	-22.5
14	-22.5
15	-1.5
16	2.5
17	-27.5
18	-31.5
19	-9.5
20	26.5
21	50.5
22	58.5
23	56.5

i	*Pre-emphasize* ke i
0	-14.5
1	-3.565
2	-2.865
3	4.685
4	-9.685
5	4.015
6	2.135
7	-2.805
8	-6.895
9	6.895
10	-10.105
11	0.595
12	-0.375
13	-10.375
14	-10.375
15	20.325
16	3.955
17	-26.045
18	-4.825
19	21.055
20	35.715
21	24.795
22	9.515
23	-0.245

2.4 Pre- emphasize

Persamaan untuk melakukan proses *Pre-emphasize* adalah sebagai berikut.

$$y\,[n] = s\,[n] - \alpha\, s[n-1]$$

Keterangan :

Y [n] = signal hasil *Pre-Emphasize filter*

S [n] = signal sebelum *Pre- Emphasize Filter*

α = 0,97

Sampel signal :

Y0 = -14.5

Y1 = (-10.5)-(-14.5 * 0.97) = 3.565

Berikut adalah datahasil *Pre-Emphasize*

Tabel 3 Data *Pre-emphasize*

2.5 Frame Blocking

Berikut adalah **p**ersamaan untuk melakukan proses *Frame Blocking.*

((I-N)/M) +1

Keterangan :

I = sample rate

N = sample point (sample rate * waktu framing(S))

M = N/2

Dengan waktu = 50 ms , dan *sampel rate* = 16.000 Hz,

maka didapatkan *Frame size* (N)

N = 16.000 * 0,05 = 800 *Sample Point*

Dan M = 400

Sehingga Jumlah *Frame* :

= ((16.000-800)/ 400))+1 = 39

2.6 *Windowing*

Berikut ini adalah persamaan yang digunakan dalam proses *windowing*.

$$w(n) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi n}{N-1}\right)$$

Keterangan:

N = jumlah sampel pada setiap *frame*

n = indeks ke-…

w = nilai *hamming window* ke-n

x = nilai sampel *signal* hasil *windowing*

Dengan menggunakan rumus $w(n) = 0{,}54 - 0.46\,cos\frac{2\pi n}{N-1}$ diperoleh hasil sebagai berikut :

Wo = 0.54 - 0.46 *cos* $\left(\frac{2*3.14*0}{800-1}\right)$= 0.08

Sehingga:

Xo = -14.5 * 0.08 = -1.16

Hasil *Windowing* dapat dilihat pada tabel 4 berikut.

Tabel 4 Data *Windowing*

i	Windowing ke i
0	-1.16
1	-0.2658
2	-0.2852
3	0.3748
4	-0.7748
5	0.3212
6	0.1708
7	-0.2244
8	-0.5516
9	0.5516
10	-0.08084
11	0.0476
12	-0.03
13	-0.83
14	-0.83
15	1.626
16	0.3164
17	-2.0836
18	-0.386
19	1.6844
20	2.8572
21	1.9836
22	0.7612
23	-0.0122

2.7 *Fast Fourier Transform* (FFT)

Berikut ini adalah persamaan proses FFT untuk mengimplementasikan *Discrete Fouries Transform* (DCT).

$$F_{[k]} = \sum_{n=1}^{N} f(n) \cos\left(\frac{2\pi nkT}{N}\right) - j \sum_{n=1}^{N} f(n) \sin\left(\frac{2\pi nkT}{N}\right)$$

Keterangan :
f(n) = sinyal suara
N = jumlah sampel
F[k] = hasil dari FFT
Untuk F0 , maka diperoleh perhitungan sebagai berikut :

$$(f_0) = \frac{1}{24}\left[-1.16 \left(cos\left(\frac{2\pi * 0 * 0}{24}\right)\right)\right] - j\, sin\, \frac{2\pi * 0 * 0}{24} +$$

$$\left[-0.2852 \left(cos\left(\frac{2\pi*0*1}{24}\right)\right)\right] - j\, sin\, \frac{2\pi*0*1}{24} +$$

$$\ldots + \left[0.0122 \left(cos\left(\frac{2\pi*0*23}{24}\right)\right)\right] - j\, sin\, \frac{2\pi*0*23}{24} = 0.0248$$

Maka F0 = 0.0248
Dengan cara yang sama lakukan perhitungan data sinyal lainnya. Sehingga akan diperoleh data sinyal hasil FFT dapat dilihat pada tabel 5 berikut.

Tabel 5 Data FFT

i	FFT ke i
0	0.0248
1	-0.0032
2	0.0045
3	0.0058
4	0.0061
5	-0.0033
6	0.0052
7	0.0078
8	0.0064

2.8 *Mel frequency Warping*

Berikut adalah rumus untuk melakukan proses *Mel Frequency Warping.*

$$Y[i] = \sum_{j=1}^{N} S[j] H_i[j]$$

Keterangan:
N = jumlah *magnitude spectrum*
S[j] = *magnitude spectrum* pada frekuensi
H_ij = *koefisien filterbank* pada frekuensi j($1 \leq i \leq M$)
M= jumlah *Channel* dalam *filterbank*
Untuk mendapatkan Hi digunakan rumus :

$$H_i = \frac{2595 * \log\left(1 + \frac{f}{700}\right)}{\frac{Si}{2}}$$

Diketahui :
S_0 = 0.0248

$$H_0 = \frac{2595 * \log\left(1 + \frac{16000}{700}\right)}{\frac{0.0248}{2}} = 228299.9769$$

Maka didapat S0 = 0.0248*228299.9769 = 7149.8394
Dengan cara yang sama maka didapatkan sinyal hasil *filterbank* dapat dilihat pada tabel 6 berikut.

Tabel 6 Data *Filterbank*

2.9

i	*Filterbank* ke i
0	7149.8394
1	6712.7077
2	6018.2896
3	5722.30822
4	6018.2896
5	5454.0750
6	6018.2896
7	5722.30822
8	6712.7077
9	7272.1003
10	7756.9067
11	10577.6006
12	10908.1506
13	7756.9067
14	7272.1003
15	6712.7077
16	38784.5335
17	13425.4154
18	174530.5003
19	34906.0801
20	5454.0750
21	10577.6004
22	10908.1506
23	7933.2006

Discrete Cosine Transform (DCT)

DCT merupakan langkah terakhir dari proses utama MFCC feature extraction. Konsep dasar dari DCT adalah mendekorelasikan mel spectrum sehingga menghasilkan representasi yang baik dari property spektral local.

Berikut adalah formula yang digunakan untuk menghitung DCT.

$$\tau_n = \sum_{k=1}^{K} (\log S_k) \cos[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}]$$

Keterangan:

S_k= keluaran dari proses *filterbank* pada indeks k

K = jumlah koefisien yang diharapkan

Koef (k) = 24

Untuk n=o maka,

$$n_{0} = \log(7149.8394)\cos\left[0\left(0-\frac{1}{2}\right)\frac{3,14}{24}\right]+$$

$$\log(6712.7077)\cos\left[0\left(1-\frac{1}{2}\right)\frac{3.14}{24}\right]+\cdots$$
$$+\log(7933.2006)\cos\left[0\left(24-\frac{1}{2}\right)\frac{3.14}{24}\right]$$

= 17.4437

Jadi no =17.4437

Begitu juga dengan data lainnya setelah dilakukan perhitungan, maka sinyal hasil *DCT* dapat dilihat pada tabel 7 berikut.

i	*DCT* ke i
0	17.4437
1	17.41414
2	17.38534
3	17.3572
4	17.32965
5	16.67375
6	16.64597
7	16.59144
8	17.10545
9	17.03262
10	17.00841
11	16.95997
12	16.9842
13	16.91138
14	16.76407
15	16.7888
16	16.71437
17	16.68942
18	16.66441
19	16.88701
20	16.83805
21	16.76407
22	16.81347
23	16.68942

Tabel 7 Data DCT

2.10 Analisis Proses Pelatihan Naive Bayesian

Pelatihan naive bayesian dilakukan untuk mendapatkan nilai mean dan varian. Berikut penjelasan dari proses pelatihan naive bayesian :

2.11 Latih Data

Latih data merupakan tahap untuk pelatihan data dengan menggunakan metode naive bayesian. Dari fitur ciri yang ada pada tabel 3.10, kemudian akan

dihitung mean dan varians setiap kelasnya, nilai mean dan varian tersebut nantinya akan digunakan untuk pencarian probabillitas likehood, dengan rumus densitas. Nilai mean dapat dihitung dengan menggunakan persamaan 2.8 dan nilai varians dapat dihitung dengan menggunakan persamaan :

$$\mu = \frac{\sum x_i}{n}$$

Dimana:
μ = mean
n = banyaknya data
$\sum x_i$ = jumlah nilai data

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

Dimana:
σ^2= varians
μ= mean
x_i = nilai data
n = banyaknya data

Berikut perhitungan nilai mean dan varians berdasarkan fitur MFCC pada tabel 3.10:

Nada Gitar G

$$\mu\ FFT\ \text{Nada G} = \frac{0{,}70463 + 0{,}68916}{2}$$

= 0,69690

$$\sigma^2\ FFT\ \text{Nada G} = \frac{1}{2-1}.(0{,}70463 - 0{,}69690)^2 + (0{,}68916 - 0{,}69690)^2$$

= 0,00012

Nada gitar A

$$\mu\ FFT\ \text{Nada A} = \frac{0{,}66979 + 0{,}64519}{2}$$

= 0,65749

$$\sigma^2\ FFT\ \text{Nada A} = \frac{1}{2-1}.(0{,}66979 - 0{,}65749)^2 + (0{,}64519 - 0{,}65749)^2$$

= 0,00030

Tabel 8 Mean dan Varian Nada G dan A

Nada	Nilai	Fitur		
		FFT	MEL FILTER	DCT
G	Mean	0.69690	4.33926	39.39922
	Varian	0.00012	0.00731	0.41231
A	Mean	0.65749	5.55330	40.79819
	Varian	0.00030	0.26281	1.16603

2.12 Simpan Data Latih

Simpan data merupakan tahap untuk menyimpan mean dan varian dari proses latih data kedalam bentuk txt.

2.13 Analisis Proses Pengujian Naive Bayesian

Pengujian dilakukan untuk mengklasifikasi akor gitar yang dimasukan dengan data yang sudah dilatih, sehingga dapat diketahui hasil klasifikasinya.

2.14 Nada Uji

Nada uji merupakan nada yang akan melalui tahap pengujian terhadap data latih, sebelumnya dilakukan tahap yang sama dengan nada latih untuk mendapatkan nilai Fitur Ciri. Contoh nada yang akan diuji dapat dilihat pada tabel 9.

Tabel 9 Contoh Nada Uji

Fitur	Nilai
FFT	0.68916
MEL FILTER	4.27881
DCT	38.94517

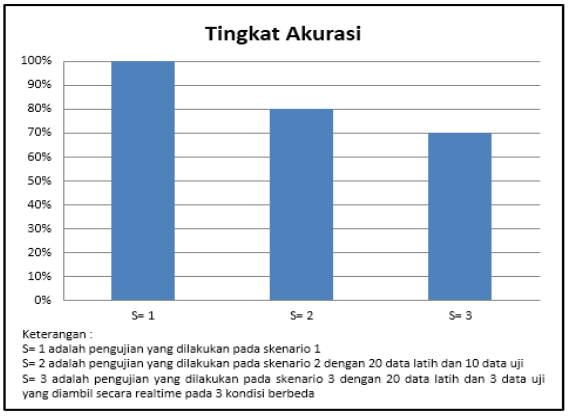
Posterior Nada G
Posterior = P(Nada G) . P(FFT | Nada G) . P(MEL | Nada G). P(DCT |
Nada G). / Evidence
Posterior = (0.5 * 28.41000 * 3.63556 * 0.48399) / 25.00132
Posterior = 24.99476 / 25.00132
Posterior = 0.99973
Posterior Nada A
Posterior = P(Nada A) . P(FFT | Nada A) . P(MEL | Nada A). P(DCT |
Nada A). / Evidence
Posterior = (0.5 * 4.37323 * 0.03541 * 0.08477) / 25.00132
Posterior = 0.00656/ 25.00132
Posterior = 0.00026
Posterior (Nada G) > posterior (Nada A).Maka kelas yang tepat untuk data uji adalah Nada G.

3. PENUTUP

3.1 Pengujian dan Hasil

Pengujian metode bertujuan untuk mengetahui hasil dari metode MFCC dan naive bayesian dalam mengklasifikasi akor gitar. Pada penelitian ini, rencana pengujian dilakukan dengan beberapa skenario. Berikut adalah skenario yang akan dilakukan.

1. Menguji nada akor gitar yang telah dijadikan data latih.
2. Menguji nada akor gitar yang belum dilatih sebelumnya dan menguji pengaruh jumlah data latih terhadap tingkat akurasi.



Berdasarkan pengujian yang telah dilakukan metode MFCC dan metode naïve Bayesian dapat digunakan untuk mengklasifikasi nada dasar gitar. dan berdasarkan pengujian diatas didapat kesimpulan bahwa konsistensi data latih berpengaruh terhadap tingkat akurasi dimana saat pengujian menggunakan instrumen yang berbeda tingkat akurasi yang dihasilkan cenderung berubah.

3.2 Kesimpulan

Hasil yang didapat dari penelitian yang telah dilakukan dalam penyusunan skripsi ini serta mengacu pada tujuan penelitian, maka dapat disimpulkan.

1. Metode MFCC dan naive bayesian dapat digunakan untuk mengklasifikasi nada dasar gitar berdasarkan suara.
2. Tingkat akurasi metode MFCC dan naive bayesian dalam mengklasifikasi nada dasar gitar berdasarkan suara adalah sebesar 70%.

3.3 Saran

Berdasarkan hasil dari penelitian yang telah tercapai saat ini, terdapat beberapa saran yang mungkin bermanfaat jika ada yang akan melakukan penelitian yang sejenis, yaitu :

1. Dataset nada yang digunakan sebaiknya memiliki kualitas yang sama.
2. Untuk mendapat tingkat keakurasian yang tinggi dalam mengklasifikasikan berbagai akor, sebaiknya menggunakan instrumen yang berkualitas baik.

DAFTAR PUSTAKA

[1] Tito Galit Permana, Dr. Ir. Bambang Hidayat, dan Eko Susatio S.T,M.T., 2013."Identifikasi Akor Gitar Menggunakan Algoritma Harmonic Product Spectrum"., Skripsi. Fakultas Elektro dan Komunikasi Universitas Telkom.

[2] Ardi Priagung, Siti Kurniawati F.,Hendri A., I Dwi Gede A., dan Oscar Sampars T., 2014., "Klasifikasi Genre Pada Lagu Dengan metode Naive Bayesian".Skripsi. Universitas Brawijaya, Malang.

[3] Eko Riyanto, Sutejo, 2014., "Perbandingan Metode Ekstraksi Ciri Suara MFCC,ZCPA DAN LPC". Skripsi. STMIK HIMSYA.

[4] Setiawan, A., Handayani, P. K. 2012. "Klastering Suara Berdasarkan Gender dengan Ekstraksi Ciri Berbasis Domain Waktu".Volume 3, No.5, Januari 2012.

[5] Sijabat, D. W. 2009. "Simulasi Pengenalan Chord Terisolasi Berbasiskan Speaker Dependent dengan Metode Hidden Markov Model". *Skripsi*.Universitas Indonesia.

[6 Anna Dara Andriana, 2013, "Perangkat Lunak Untuk Membuka Aplikasi Pada Komputer Dengan Perintah Suara Menggunakan Metode *Mel Frequency Cepstrum Coefficient".* Volume 2, No.1, Maret 2013.

[7] Huda, M., Basuki, D. K., Akbar, F., Permana, F. J. 2011. "Konversi Nada-Nada Akustik Menjadi Chord Menggunakan Pitch Class Profile". *Skripsi*.Institut Teknologi Sepuluh November.

[8] Wisnudisastra E. 2009. "Pengenalan Chord Pada Alat Musik Gitar Menggunakan Codebook dengan Teknik Ekstraksi Ciri MFCC". *Skripsi*. Institut Pertanian Bogor.

[9] Maulana, M. A. 2013. "Implementasi Speech Recognition pada Permainan Toddler Card untuk Anak Usia Toddler Berbasis Android". *Skripsi*. Universitas Sumatera Utara.