

# Klasifikasi Gamelan Bali Menggunakan Support Vector Machine (SVM)

I Kadek Riski Ari Putra<sup>a1</sup>, Luh Arida Ayu Rahning Putri<sup>a2</sup>, I Made Widiartha<sup>a3</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana  
Badung, Bali, Indonesia

<sup>1</sup>fxdeki@gmail.com

<sup>2</sup>rahningputri@unud.ac.id

<sup>3</sup>madewidiartha@unud.ac.id

## Abstract

*Balinese gamelan is traditional Balinese music in the form of an orchestra which consists of various kinds of instruments, using pelog and celendro scales. Various aspects of life and communal activities in Bali cannot be separated from gamelan religious ceremonies, official ceremonies, to entertainment. For ordinary people and tourists, distinguishing Balinese gamelan is not an easy thing, because not everyone has the knowledge of Balinese gamelan, as well as the experience of hearing and seeing Balinese gamelan. This research aims to classify Balinese gamelan based on their functions into two classes, namely Ceremonial Gamelan and Gamelan for entertainment using Support Vector Machine (SVM), as well as Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction from audio. The pre-processing stages include normalization and pre-emphasis, the features used are 13 MFCC coefficients, the classification process uses SVM with the RBF kernel. The results obtained is an SVM classification model with 86% accuracy using the parameters  $C=5$ ,  $\gamma=0.1$ , and  $\text{tol}=1e-1$ .*

**Keywords:** Bali, Gamelan, Mel-Frequency Cepstral Coefficients, Support Vector Machine, Classification, RBF.

## 1. Pendahuluan

Gamelan Bali merupakan musik tradisional Bali berupa orkestra yang terdiri dari berbagai macam instrumen. Gamelan Bali pada dasarnya menggunakan laras atau tangga nada bertipe *pelog* dan *selendro*. Berbagai aspek kehidupan dan kegiatan masyarakat di Bali tidak dapat terlepas dari gamelan mulai dari upacara keagamaan, upacara resmi, hingga hiburan. Sehingga dapat dikatakan bahwa hampir seluruh kegiatan di Bali mengikutsertakan gamelan di dalamnya. Bagi masyarakat yang awam maupun wisatawan membedakan gamelan Bali bukanlah hal yang mudah, hal ini disebabkan pengetahuan mengenai gamelan Bali yang tidak mendalam, serta pengalaman mendengar, dan melihat permainan gamelan Bali tidak dimiliki oleh semua orang. Hanya kalangan tertentu saja yang mengetahui penggolongan gamelan Bali seperti pecinta seni gamelan Bali itu sendiri, peneliti, dan para seniman pemain gamelan Bali [1].

Banyak masyarakat maupun wisatawan yang tertarik terhadap budaya gamelan Bali, namun membedakan gamelan Bali bukanlah hal yang mudah sebab pengetahuan mengenai gamelan Bali yang tidak dimiliki semua orang. Oleh karena itu perlu diciptakan sistem yang dapat membantu untuk mengklasifikasikan gamelan Bali secara otomatis berdasarkan fungsi gamelan tersebut, sehingga masyarakat awam dapat mengetahui fungsi gamelan tersebut dan dapat merepresentasikan serta menikmati gamelan tersebut dengan baik.

Penelitian ini membahas mengenai klasifikasi gamelan Bali ke dalam dua kelas yaitu gamelan upacara dan gamelan hiburan menggunakan *Support Vector Machine* (SVM) dan *Mel-Frequency Cepstral Coefficients* (MFCC) sebagai metode ekstraksi fitur dari gamelan. Pemilihan MFCC sebagai metode ekstraksi fitur didasarkan pada penelitian terdahulu [2][3][4] yang mendapatkan hasil yang baik dari ekstraksi fitur MFCC pada data audio terkait musik. Di mana MFCC dapat merepresentasikan sinyal audio dengan baik dan memiliki cara kerja yang didasarkan pada frekuensi yang sesuai dengan

pendengaran manusia [5]. Metode SVM digunakan dalam klasifikasi karena berdasarkan penelitian terdahulu, kombinasi MFCC dan SVM mendapatkan hasil yang baik dalam melakukan klasifikasi audio [2][3][6]. SVM juga memenuhi salah satu kebutuhan utama machine learning yaitu kecepatan dalam pelatihan model [7].

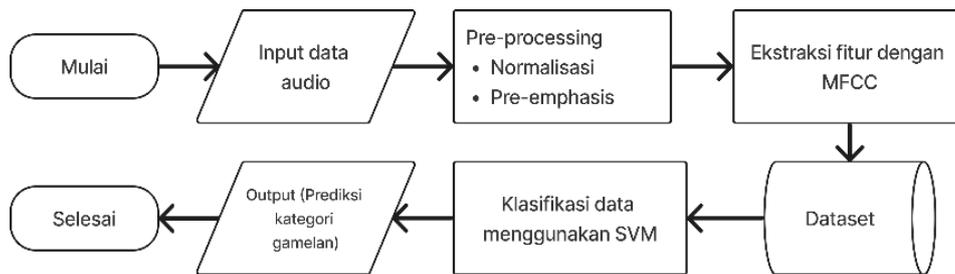
## 2. Metodologi Penelitian

### 2.1. Data Penelitian

Data yang akan digunakan pada penelitian merupakan data primer yang berupa *file* audio gamelan dari seorang pakar gamelan Bali yang telah diberi label. Data akan diubah menjadi *file* .wav, kemudian dipotong pada bagian yang perbedaannya paling menonjol yaitu bagian *refrain* menggunakan tools Audacity sepanjang 30 detik. Total data berjumlah 100 *file* audio dari gamelan yang berbeda-beda, dengan jumlah 50 audio gamelan upacara dan 50 audio gamelan hiburan.

### 2.2. Desain Sistem

Penelitian ini mengimplementasikan metode *Support Vector Machine* (SVM) untuk klasifikasi gamelan Bali. *Input* yang akan diterima oleh sistem berupa *file* audio yang akan melalui proses *pre-processing*, ekstraksi fitur, dan klasifikasi.



**Gambar 1.** Gambaran Alur Kerja Sistem

Seperti pada grafik yang dapat dilihat pada Gambar 1, alur kerja dari sistem yang akan dibangun adalah sebagai berikut.

#### a. *Input Data Audio*

Sistem akan menerima *input* berupa *file* audio berbentuk .wav dengan durasi 30 detik. Dengan bantuan *library python librosa.load* *file* audio akan dimuat menjadi sebuah array yang berisi amplitudo dari sinyal audio di mana setiap nilai dalam array tersebut mewakili amplitudo dari audio pada waktu tertentu. *Sample rate* (jumlah sampel audio per detik) yang digunakan pada penelitian ini adalah 44100 sampel per detik (44,1 kHz).

#### b. *Pre-processing Data*

Sebelum data melalui proses ekstraksi fitur akan dilakukan *pre-processing* data terlebih dahulu untuk meningkatkan kualitas dari data sebelum diolah lebih lanjut dalam penelitian. Adapun proses *pre-processing* data yang dilakukan dalam penelitian ini adalah sebagai berikut:

##### 1. Normalisasi

Normalisasi pada penelitian ini dilakukan dengan *min-max normalization* dengan persamaan (1) yang akan mentransformasi nilai terendah dalam data ke 0 dan nilai tertinggi dalam data ke 1 sehingga nilai-nilai di antaranya terdistribusi secara proporsional.

$$norm(y) = \frac{y - \min(y)}{\max(y) - \min(y)} \quad (1)$$

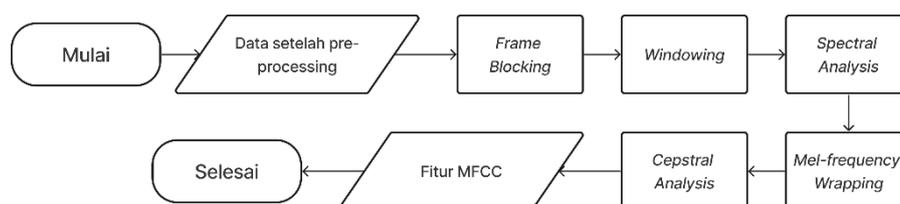
##### 2. *Pre-emphasis*

Proses pre-emphasis dengan persamaan (2) merupakan penekanan pada frekuensi tinggi untuk menangkap detail yang ada pada frekuensi tersebut. Dalam penelitian ini koefisien yang digunakan dalam pre-emphasis adalah 0.95.

$$y[n] = x[n] - coef \times x[n - 1] \quad (2)$$

### c. Ekstraksi Fitur Menggunakan MFCC

Data yang telah diolah menjadi potongan audio gamelan Bali selanjutnya akan diekstraksi menggunakan *Mel-Frequency Cepstral Coefficient*. MFCC merupakan salah satu metode yang sering digunakan untuk ekstraksi fitur karena memiliki waktu ekstraksi yang lebih cepat dan akurasi yang lebih tinggi dibandingkan dengan metode lainnya [8]. Ekstraksi fitur MFCC akan dilakukan menggunakan bahasa *python* untuk mendapatkan fitur-fitur dari data audio. Langkah-langkah yang dilakukan dalam ekstraksi fitur diilustrasikan pada Gambar 2 [4].



**Gambar 2.** Ekstraksi Fitur MFCC

Berdasarkan grafik yang dapat dilihat pada Gambar 2, berikut merupakan proses ekstraksi fitur MFCC.

#### 1. *Input* Proses MFCC

Sistem akan menerima data berupa hasil dari proses *pre-processing* sebelumnya yang meliputi normalisasi dan *pre-emphasis*.

#### 2. *Frame Blocking*

Dalam proses ini sinyal suara akan disegmentasi menjadi beberapa *frame* berukuran 40ms dengan *overlapping* (tumpukkan) sebesar 50% mengacu pada [9] yang menyebutkan bahwa *frame* diambil sepanjang 40ms untuk mendapatkan resolusi frekuensi yang baik.

#### 3. *Windowing*

Proses *windowing* dilakukan untuk mengurangi diskontinuitas sinyal setelah *frame blocking*. Proses ini dilakukan dengan fungsi *hamming window*. *Hamming window* dapat menghasilkan *sidelobe* (kebocoran) yang tidak terlalu tinggi dan *noise* yang tidak terlalu besar sehingga menjadi fungsi *window* yang paling sering digunakan [10].

#### 4. *Spectral Analysis*

Proses *spectral analysis* dilakukan dengan *Fast Fourier Transform*, setiap *frame* akan diubah menjadi domain frekuensi untuk kemudian dipetakan menjadi *mel-frequency*.

#### 5. *Mel-frequency wrapping*

Dalam proses ini, frekuensi akan diubah ke spektrum mel dalam skala linear untuk frekuensi di bawah 1000 Hz dan skala logaritmik untuk frekuensi di atas 100 Hz.

#### 6. *Cepstral Analysis*

Proses *cepstral analysis* dilakukan dengan *Cepstrum* yang bertujuan untuk menghasilkan fitur dari ekstraksi yang telah dilakukan. Data diubah dari domain frekuensi menjadi domain waktu kembali yang kemudian akan menghasilkan matriks fitur MFCC dengan koefisien MFCC sejumlah 13 koefisien.

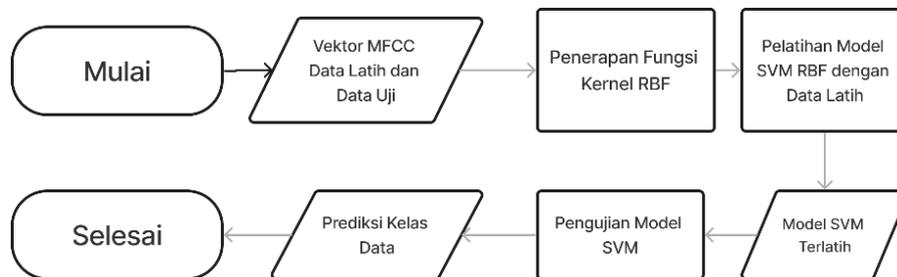
## 7. Output MFCC

Hasil dari proses ekstraksi MFCC berupa matriks MFCC yang tersusun dari frame dan koefisien MFCC yang memiliki ukuran jumlah frame  $\times$  jumlah koefisien MFCC. Jumlah koefisien MFCC yang digunakan adalah sebanyak 13, hal ini dikarenakan koefisien MFCC di awal sering kali menangkap karakteristik spektral musik yang paling relevan karena MFCC mengadopsi cara kerja telinga manusia yang lebih sensitif terhadap perubahan suara di frekuensi rendah, serta untuk mengurangi dimensi fitur sehingga dapat meningkatkan efisiensi komputasi. Pada penelitian terdahulu 13 koefisien MFCC dapat memberikan akurasi klasifikasi hingga 91% saat menggunakan SVM dengan *kernel Gaussian* (RBF) [6].

Hasil MFCC dari setiap data (1 *file audio*) akan disimpan dalam bentuk array 1 dimensi dengan mengambil nilai rata-rata (*means*) dari semua frame MFCC0 sampai MFCC12. Sehingga setiap array akan memiliki panjang 13 sesuai dengan jumlah koefisien MFCC yang diekstraksi dan setiap koefisien MFCC akan memiliki 1 nilai saja yang merepresentasikan fitur spektral dari keseluruhan *frame* dalam 1 data (*file audio*). Dataset akan berdimensi  $100 \times 13$  yang tersusun dari jumlah keseluruhan data dan jumlah koefisien MFCC yang diekstraksi.

### d. Klasifikasi Dengan SVM

Tahap klasifikasi dengan SVM merupakan tahap pembangunan model dengan metode SVM agar dapat mengklasifikasikan gamelan Bali. Pada penelitian ini pelatihan akan dilakukan menggunakan *kernel radial basis function* (RBF). Kernel RBF dipilih karena memiliki kinerja yang baik dalam SVM untuk melakukan klasifikasi audio. Penelitian terdahulu untuk klasifikasi audio memberikan hasil yang baik dengan menggunakan kernel RBF [2], [6]. Proses pelatihan dan pengujian SVM diilustrasikan pada Gambar 3 [11].



**Gambar 3.** Klasifikasi Dengan SVM

Pada grafik dalam Gambar 3 dapat dilihat bahwa proses klasifikasi menggunakan SVM adalah sebagai berikut.

#### 1. Input Data

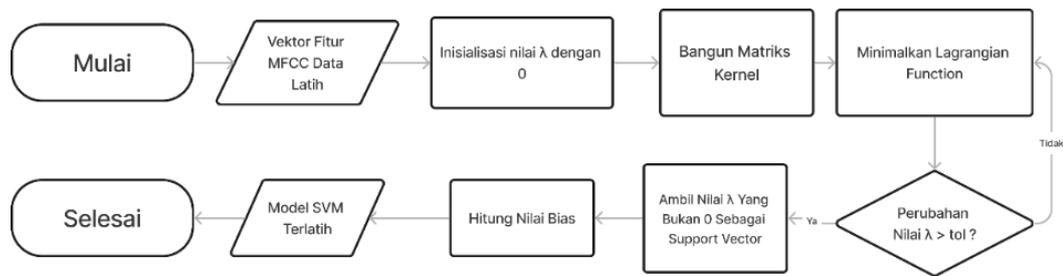
Data yang akan diterima adalah vektor fitur MFCC dari hasil proses ekstraksi fitur yang terbagi menjadi dua yaitu data latih dan data uji dengan rasio perbandingan 80:20. Data latih akan digunakan pada Pelatihan Model dan data uji akan digunakan pada Pengujian Model SVM Terlatih seperti pada Gambar 3.

#### 2. Penerapan Fungsi *Kernel* RBF

Pada algoritma SVM akan diterapkan *kernel radial basis function* yang mengklasifikasikan data non-linear dengan menghitung jarak antar data dalam dimensi yang lebih tinggi dengan persamaan (4).

#### 3. Pelatihan Model SVM

Pada proses ini Model SVM akan dilatih untuk mendapatkan *hyperplane* optimal dan terbaik untuk mengklasifikasikan data. Proses pelatihan model diilustrasikan dalam *flowchart* pada Gambar 4 [12].



Gambar 4. Pelatihan Model SVM

- Inisialisasi nilai  $\lambda$  dengan 0 agar memenuhi syarat pada persamaan (1) dan (2)

$$0 \leq \lambda_i \leq C \quad (1)$$

$$\sum_i \lambda_i = 0 \quad (2)$$

- Bangun matriks kernel (3) dengan menggunakan fungsi kernel pada persamaan (4).

$$K = \begin{pmatrix} y_1 y_1 K(x_1; x_1) & y_1 y_2 K(x_1; x_2) & \dots & y_1 y_N K(x_1; x_N) \\ y_2 y_1 K(x_2; x_1) & y_2 y_2 K(x_2; x_2) & \dots & y_2 y_N K(x_2; x_N) \\ \dots & \dots & \dots & \dots \\ y_N y_1 K(x_N; x_1) & y_N y_2 K(x_N; x_2) & \dots & y_N y_N K(x_N; x_N) \end{pmatrix} \quad (3)$$

$$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2} \quad (4)$$

- Minimalkan lagrangian function dengan persamaan (5) hingga perubahan nilainya lebih kecil dari parameter **tol** yang merupakan toleransi perubahan  $\lambda$  antar iterasi.

$$L^* = k_0^T v_0 + \frac{1}{2} v_0^T Q v_0 \quad (5)$$

$$v_0 = (\lambda_M, \lambda_L)^T \quad (6)$$

$$k_0 = (1 - \lambda^T K_M, 1 - \lambda^T K_L)^T \quad (7)$$

$$Q = \begin{pmatrix} K_{M,M} & K_{M,L} \\ K_{L,M} & K_{L,L} \end{pmatrix} \quad (8)$$

Di mana

$\lambda_i$  merupakan nilai bobot setiap titik data

$x$  merupakan nilai fitur data

$y$  merupakan label dari data

$K$  merupakan fungsi kernel

$C$  merupakan *penalty parameter* yang mengatur toleransi terhadap misklasifikasi

- Ambil nilai  $\lambda$  yang bukan nol sebagai *support vector*.
- Hitung nilai *bias* dengan persamaan (9).

$$b = \mathbb{E}_k [y_k - \sum_i \lambda_i y_i (x_i \cdot x_k) | \lambda_k > 0] \quad (9)$$

Di mana

$\lambda_i$  merupakan nilai bobot setiap titik data

$x_i$  merupakan vektor fitur titik data ke- $i$

$x_k$  merupakan vektor fitur *support vector* ke- $k$

$y_i$  merupakan label dari titik data ke- $i$

$y_k$  merupakan label dari *support vector* ke- $k$

$(x_i \cdot x_k)$  merupakan *dot product* yang didapatkan dari fungsi kernel

Setelah melalui proses pelatihan, maka model SVM dapat digunakan untuk melakukan prediksi/klasifikasi.

d. Pengujian Model SVM

Pengujian pada Model SVM ini bertujuan untuk mengetahui apakah model dapat melakukan prediksi atau klasifikasi dengan baik. Pengujian dilakukan dengan menggunakan model untuk melakukan prediksi kelas pada sebuah data dengan persamaan (10).

$$class = sign(f(x)), \quad f(x) = \sum_i \lambda_i y_i (x_i \cdot x) + b \quad (10)$$

Di mana

$\lambda_i$  merupakan nilai bobot setiap titik data

$x_i$  merupakan vektor fitur titik data ke-i

$x_k$  merupakan vektor fitur *support vector* ke-k

$y_i$  merupakan label dari titik data ke-i

$y_k$  merupakan label dari *support vector* ke-k

$(x_i \cdot x_k)$  merupakan *dot product* yang didapatkan dari fungsi kernel

e. *Output* (Hasil Klasifikasi)

Setelah proses pelatihan model selesai maka model dapat digunakan untuk melakukan klasifikasi/prediksi dan akan menampilkan hasil klasifikasi tersebut yang berupa prediksi kelas dari data gamelan Bali yang diberikan pada model.

### 2.3 Desain Evaluasi

Model SVM yang dikembangkan akan diuji menggunakan K-Fold Cross Validation. K-Fold Cross Validation merupakan metode yang digunakan untuk menguji rata-rata keberhasilan suatu sistem [13]. K-Fold Cross Validation bekerja dengan melakukan perulangan yang akan mengacak data menjadi k fold dan menggunakan k-1 fold sebagai data latih dan 1 fold sebagai data uji secara bergantian. Dalam penelitian ini, jumlah fold (K) yang digunakan adalah 10 fold.

Untuk memperoleh model yang optimal maka dilakukan *hyperparameter tuning* pada model SVM. Terdapat 3 *hyperparameter* yang akan diuji coba yaitu *penalty parameter* (C) dan *kernel coefficients* (gamma) serta nilai toleransi perubahan  $\lambda$  antar iterasi (tol) pada model SVM.

Nilai C (*penalty parameter*) merupakan parameter yang menentukan toleransi terhadap misklasifikasi di mana semakin besar nilai C maka toleransi terhadap misklasifikasi akan lebih kecil sehingga SVM akan mencoba mencari *hyperplane* terbaik dengan menghindari misklasifikasi. Rentang nilai C yang akan diujikan adalah 1, 5, 10, 15 dan 20. Semakin besar nilai C maka *hyperplane* akan memiliki margin yang lebih kecil, dan jika *hyperplane* tersebut dapat bekerja dengan baik maka model SVM akan semakin spesifik. Namun, jika nilai C terlalu besar maka hal ini dapat menyebabkan *overfitting* [14].

Nilai gamma (*kernel coefficients*) menentukan seberapa jauh pengaruh dari sebuah titik data dalam pembentukan *hyperplane* yang memisahkan antara dua kelas, nilai gamma yang akan diujikan adalah 0.05, 0.075, 0,1, 0.125 dan 0.15 dengan mempertimbangkan jarak data terhadap keterkaitan antar data, di mana dengan nilai gamma kecil keterkaitan antar data bisa saja terjadi meski jarak antar data tersebut jauh. Sedangkan, dengan nilai gamma yang lebih besar jarak antar data yang jauh akan memperkecil keterkaitannya.

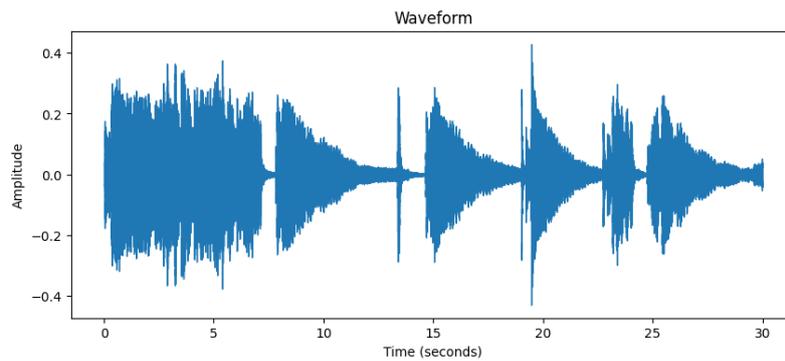
Nilai tol (toleransi perubahan  $\lambda$  antar iterasi) merupakan pemeriksaan perubahan nilai  $\lambda$  saat pelatihan model SVM. Hal ini dilakukan agar penggunaan sumber daya komputasi lebih efektif sehingga pelatihan akan dihentikan saat perubahan nilai  $\lambda$  sudah lebih kecil dari nilai toleransi. Beberapa nilai toleransi pelatihan yang akan diuji coba dalam penelitian ini adalah  $1e-3$  hingga  $1e+1$ . Evaluasi ini bertujuan untuk menemukan nilai *hyperparameter* terbaik yang dapat menghasilkan akurasi tertinggi pada model SVM dalam jumlah iterasi pelatihan yang kecil.

Akurasi model SVM yang dikembangkan dihitung dengan persamaan (11). Nilai akurasi diukur berdasarkan jumlah prediksi yang sesuai atau terklasifikasikan dengan benar yang dibagi dengan jumlah data uji.

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Data\ Uji} \times 100\% \quad (11)$$

### 3. Hasil dan Diskusi

#### 3.1. Representasi Data Audio Gamelan

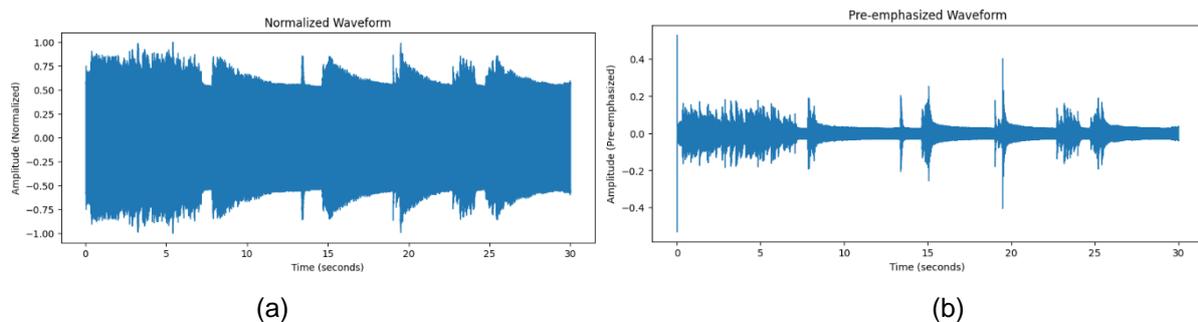


**Gambar 5.** Representasi Data Audio

Pada grafik yang dapat dilihat pada Gambar 5, grafik tersebut menampilkan amplitudo dalam gelombang suara data audio. Terdapat sejumlah 1.323.000 sampel pada setiap file audio, di mana setiap file audio berdurasi 30 detik dengan 44.100 sampel. Contoh ilustrasi hasil pembacaan file audio 'pertunjukan\_025.wav' pada Gambar 6.

#### 3.2. Pre-processing Data

Data audio gamelan kemudian masuk tahap *pre-processing* berupa normalisasi dengan min-max normalization untuk meratakan amplitudo dan pre-emphasis untuk meningkatkan efisiensi analisis sinyal audio dengan tujuan untuk meningkatkan sinyal di frekuensi tinggi untuk menangkap detail yang ada di frekuensi tersebut. Gelombang amplitudo audio setelah normalisasi dapat dilihat pada Gambar 6a dan gelombang audio setelah pre-emphasis dapat dilihat pada Gambar 6b.



**Gambar 6.** Hasil *Pre-processing* (a) Hasil Normalisasi (b) Hasil *Pre-emphasis*

#### 3.3. Ekstraksi Fitur MFCC

Setelah mendapatkan data audio yang telah melalui proses pre-processing maka dapat dilanjutkan ke proses ekstraksi fitur menggunakan Mel-Frequency Cepstral Coefficients (MFCC). Hasil Ekstraksi berupa matriks fitur MFCC sejumlah 13 koefisien dari MFCC0-MFCC12 seperti pada Tabel 1.

**Tabel 1.** Matriks Fitur MFCC

Frame	MFCC0	MFCC1	...	MFCC12
0	12.880716	-0.011487	...	0.000849
1	-4.479448	-0.401039	...	-0.220288
...	...	...	...	...
749	-5.429123	0.662139	...	-0.038364

Matriks fitur MFCC akan disimpan sebagai vektor fitur dalam bentuk tabel dataset dengan mengambil nilai rata-rata (means) dari seluruh frame untuk setiap koefisien MFCC sehingga 1 file audio akan memiliki 1 vektor fitur MFCC seperti pada Tabel 2 dimana terdapat nama *file*, 13 koefisien MFCC dari MFCC0 hingga MFCC12, dan label dari setiap data.

**Tabel 2.** Dataset Vektor Fitur MFCC

Filename	MFCC0	MFCC1	...	MFCC12	Label
angklungklentangan_002.wav	4.950594	0.360541	...	-0.674247	Upacara
angklungklentangan_003.wav	5.041135	0.783111	...	-0.566654	Upacara
...	...	...	...	...	...
pertunjukan_025.wav	1.311476	3.356117	...	-0.426246	Hiburan

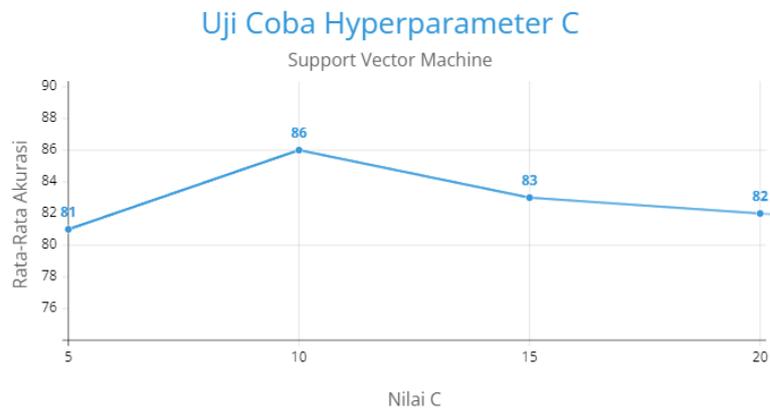
### 3.4. Klasifikasi Dengan SVM

Dataset yang dihasilkan dari proses ekstraksi fitur MFCC kemudian akan digunakan dalam pelatihan model SVM yang dilatih menggunakan *hyperparameter default* berupa  $C=1$ ,  $\gamma=0.1$ , dan  $\text{tol}=1e-5$ . Kemudian model diuji untuk mengetahui apakah model dapat melakukan prediksi atau klasifikasi dengan baik.

### 3.5. Evaluasi

Evaluasi pada model SVM yang dikembangkan berfungsi untuk mengetahui kinerja terbaik dari model tersebut. Untuk mengetahui kinerja terbaik model SVM dilakukan *K-Fold Cross Validation* sebanyak 10 *fold* serta *hyperparameter tuning* pada model SVM, sehingga didapatkan nilai akurasi SVM yang terbaik. *Hyperparameter* yang diuji coba adalah  $C$  (*penalty parameter*),  $\gamma$  (koefisien dari *kernel RBF*), dan  $\text{tol}$  (nilai toleransi perubahan  $\lambda$  antar iterasi) dengan *10 fold Cross Validation* kemudian rata-rata hasil pengujian akan digunakan sebagai nilai akhir.

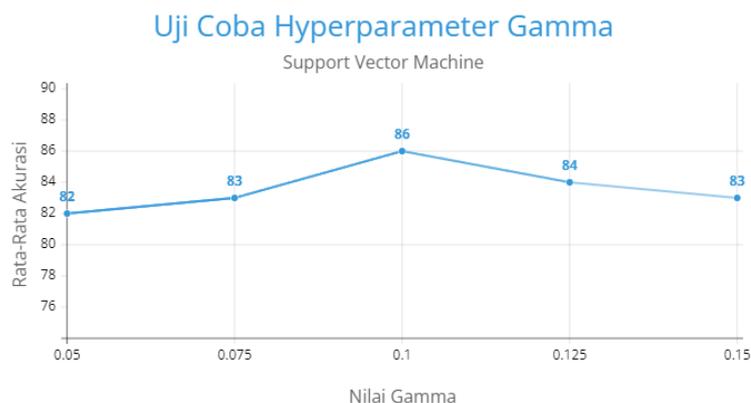
#### 1. Uji Coba *Hyperparameter C*



**Gambar 7.** Hasil Uji Coba Parameter C

Hasil uji coba *tuning* parameter C dapat dilihat pada grafik dalam Gambar 7, parameter lain yang digunakan dalam uji coba parameter C adalah  $\gamma=0.1$ , dan  $\text{tol}=1e-5$ . Gambar 7 menunjukkan bahwa perubahan nilai C dari parameter awal yaitu 1 ke 5 menunjukkan peningkatan rata-rata akurasi dari 81% ke 86%, nilai C 10 menurunkan rata-rata akurasi ke 83%, nilai C 15 menurunkan rata-rata akurasi ke 82%, dan nilai C 20 menurunkan rata-rata akurasi ke 81%. Sehingga nilai 5 merupakan nilai C terbaik yang dapat digunakan dalam penelitian ini.

#### 2. Uji Coba *Hyperparameter Gamma*



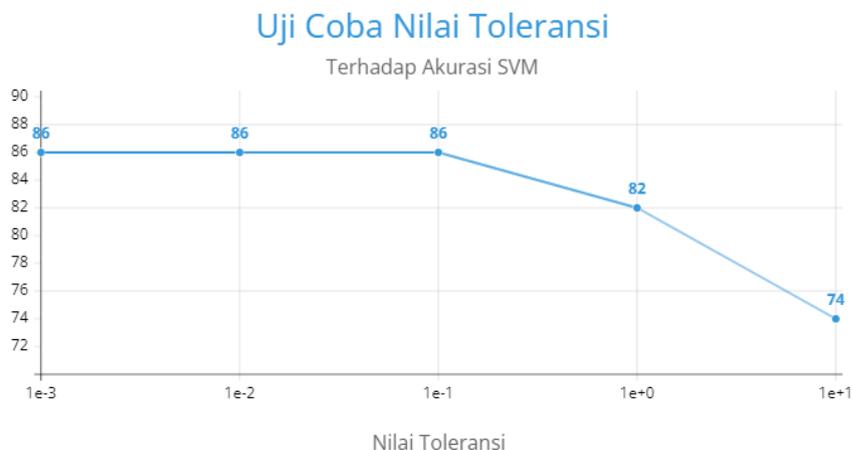
**Gambar 8.** Hasil Uji Coba Parameter Gamma

Hasil uji coba *tuning* parameter gamma dapat dilihat pada grafik dalam Gambar 8, parameter lain yang digunakan dalam uji coba parameter gamma adalah  $C=5$ , dan  $tol=1e-5$ . Gambar 8 menunjukkan bahwa nilai gamma 0.05 menghasilkan rata-rata akurasi sebesar 82%, nilai gamma 0.075 meningkatkan akurasi ke 83%, nilai gamma 0.1 meningkatkan akurasi ke 86%. Kemudian nilai gamma 0.125 menurunkan akurasi ke 84%, dan nilai gamma 0.15 menurunkan akurasi ke 83%. Sehingga, berdasarkan hasil pengujian tersebut nilai 0.1 merupakan nilai gamma terbaik yang dapat digunakan pada penelitian ini.

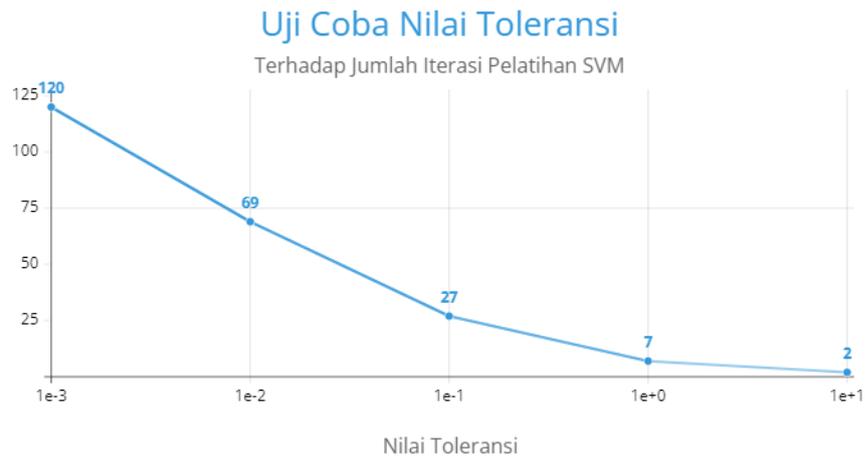
### 3. Uji Coba Nilai Toleransi

Parameter  $tol$  (nilai toleransi perubahan  $\lambda$  antar iterasi) dalam pelatihan model SVM akan berpengaruh terhadap akurasi klasifikasi model SVM dan beban komputasi selama pelatihan model SVM. Oleh karena itu, perlu ditemukan nilai toleransi pelatihan terbaik yang memiliki jumlah iterasi paling sedikit untuk mencapai akurasi terbaik dari model SVM.

Hasil uji coba nilai toleransi dapat dilihat pada grafik dalam Gambar 9, parameter lain yang digunakan dalam uji coba parameter  $tol$  adalah  $C=5$ , dan  $gamma=0.1$  yang merupakan nilai terbaik parameter tersebut. Gambar 9 menunjukkan bahwa meningkatkan nilai  $tol$  mulai dari  $1e-3$  tidak merubah rata-rata akurasi kinerja model hingga nilai  $1e-1$ . Namun, meningkatkan nilai  $tol$  ke  $1e+0$  menurunkan akurasi kinerja model ke 82%, kemudian nilai  $tol$   $1e+1$  menurunkan akurasi model hingga 74%. Grafik pada Gambar 10 menunjukkan bahwa meningkatkan nilai  $tol$  menurunkan akan jumlah iterasi pelatihan, namun mengacu pada grafik dalam Gambar 9, nilai  $1e+0$  akan menurunkan akurasi model ke 82%, dan nilai  $1e+1$  menurunkan akurasi hingga 74% meskipun menurunkan jumlah iterasi.



**Gambar 9.** Hasil Uji Coba Nilai Toleransi Terhadap Akurasi



**Gambar 10.** Hasil Uji Coba Nilai Toleransi Terhadap Jumlah Iterasi

Berdasarkan hasil pengujian tersebut nilai  $1e-1$  merupakan nilai toleransi terbaik yang dapat digunakan dalam penelitian ini karena memiliki jumlah iterasi yang paling sedikit namun dapat menghasilkan akurasi maksimal pada model SVM.

Rata-rata akurasi terbaik model adalah sebesar 86% dalam rata-rata 27 iterasi yang diperoleh dengan *hyperparameter* berupa **C=5**, **gamma=0.1**, serta **tol=1e-1**. Hal ini menunjukkan bahwa optimasi *hyperparameter* dapat meningkatkan efisiensi dan performa model.

#### 4. Kesimpulan

Berdasarkan analisis dan hasil pengujian dalam penelitian ini, dapat disimpulkan bahwa Metode ekstraksi fitur *Mel-Frequency Cepstral Coefficients* dan metode *Support Vector Machine* dapat digunakan untuk mengklasifikasikan gamelan Bali berdasarkan fungsinya ke dalam dua kelas yaitu Gamelan Upacara dan Gamelan Hiburan dengan rata-rata akurasi tertinggi sebesar 86% yang diperoleh dengan menggunakan 13 koefisien MFCC, serta model SVM dengan nilai *hyperparameter* C=5, gamma=0.1, dan toleransi perubahan  $\lambda$  antar iterasi (tol)= $1e-1$ .

#### Referensi

- [1] I. G. Harsemadi, "Purwarupa Sistem Ekstraksi Fitur Audio Pada Musik Gamelan Tradisional Bali," in *Seminar Nasional Telekomunikasi dan Informatika (SELISIK 2018)*, 2018, pp. 251–258.
- [2] A. Danika, J. Raharjo, and B. Hidayat, "Deteksi Suara Gitar Dengan Bahan Jenis Senar Berbeda Melalui Ciri Akustik Dengan Mel-Frequency Cepstral Coefficients (MFCC) Dan Support Vector Machine (SVM)," *eProceedings of Engineering*, vol. 9, no. 6, pp. 2936–2942, 2023.
- [3] S. Y. Yehezkiel and Y. Suyanto, "Music Genre Identification Using SVM and MFCC Feature Extraction," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 12, no. 2, p. 115, Oct. 2022, doi: 10.22146/ijeis.70898.
- [4] P. D. Prasetyo, I. G. P. S. Wijaya, and A. Y. Husodo, "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)," *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTika)*, vol. 1, no. 2, pp. 189–197, 2019, [Online]. Available: <http://jtika.if.unram.ac.id/index.php/JTIKA/>
- [5] Ajrana, A. Akbar, and A. Lawi, "Implementasi Algoritma Deep Artificial Neural Network Menggunakan Mel Frequency Cepstrum Coefficient Untuk Klasifikasi Audio Emosi Manusia," in *KONIK (Konferensi Nasional Ilmu Komputer)*, 2021, pp. 66–73. [Online]. Available: <https://www.kaggle.com/ejlok1/toronto->

- [6] R. Thiruvengatanadhan, "Music Classification using MFCC and SVM," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 9, pp. 922–924, 2018.
- [7] D. P. Adi and A. B. Gumelar, "Deteksi Emosi Wicara pada Media On-Demand menggunakan SVM dan LSTM," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 799–804, 2020.
- [8] K. Nugroho, E. Winarno, E. Zuliarso, and Sunardi, "Multi-Accent Speaker Detection Using Normalize Feature MFCC Neural Network Method," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 832–836, Aug. 2023, doi: 10.29207/resti.v7i4.4652.
- [9] Heriyanto, "DETEKSI SUARA UCAPAN SALAM BAHASA ARAB MENGGUNAKAN MEL FREQUENCY CEPSTRAL COEFFICIENT (MFCC) DAN PEMILIHAN FITUR MIN MAX," *Seminar Nasional Informatika (SEMNASIF)*, vol. 1, no. 1, pp. 405–414, 2020.
- [10] D. P. Candra, "KLASIFIKASI SUARA DENGAN EKSTRAKSI CIRI MEL FREQUENCY CEPSTRAL COEFFICIENTS MENGGUNAKAN MACHINE LEARNING," UIN Syarif Hidayatullah, Jakarta, 2021.
- [11] R. Adawiyah and D. I. Mulyana, "Optimasi Deteksi Penyakit Kulit Menggunakan Metode Support Vector Machine (SVM) dan Gray Level Co-occurrence Matrix (GLCM)," *Optimasi Deteksi Penyakit Kulit Menggunakan Metode Support Vector Machine (SVM) dan Gray Level Co-occurrence Matrix (GLCM)*, vol. 14, no. 1, pp. 18–33, 2022.
- [12] A. V. Khlevniuk and V. B. Tymchyshyn, "Yet more simple SMO algorithm," 2020. doi: 10.13140/RG.2.2.36670.10562.
- [13] H. Sulistiani, K. Muludi, and A. Syarif, "Implementation of Dynamic Mutual Information and Support Vector Machine for Customer Loyalty Classification," *J Phys Conf Ser*, vol. 1338, no. 1, pp. 012–050, Oct. 2019, doi: 10.1088/1742-6596/1338/1/012050.
- [14] R. G. J. Respati, "Identifikasi Emosi Melalui Suara Menggunakan Support Vector Machine Dan Convolutional Neural Network," Universitas Islam Indonesia, Yogyakarta, 2021.

*This page is intentionally left blank.*