BAB II
KAJIAN LITERATUR

2.1 Penelitian Terdahulu

Penelitian terdahulu yang relevan dengan klasifikasi lagu daerah di Indonesia ditunjukkan pada tabel 2.1.

Tabel 2. 1 Perbandingan Penelitian Terdahulu

No	Peneliti	Permasalahan	Metode	Hasil	Kelebihan	Kekurangan
1	Pujo hari Saputro et al. (2017)	Klasifikasi lagu daerah berdasarkan lirik	TF-IDF +Naive Bayes	Akurasi 72,63%	Mampu mengklasifikasi lagu berdasarkan aspek linguistik	Hanya menggunakan lirik, tidak memanfaatkan fitur audio dan karakteristik akustik lagu
2	Syafti Dhanusafitri	Klasifikasi lagu lokal daerah indonesia berdasarkan nada dan tempo	MFCC + CNN	Akurasi 63,35%	Menggunakan ektraksi fitur MFCC yang efektif untuk audio	Dataset terbatas hanya 4 kelas daerah dan belum menggunakan visualissasi interaktif
3	Gst. Ayu Vida Mastrika Giria & made Leo Radhityaa (2023)	Klasifikasi lagu daerah indonesia menggunakan *machine learning*	MFCC Spectral Features + SVM/KNN	Akurasi terbaik 73% menggunakan SVM	Menggunakan dataset IRSD dan fitur audio yang beragam	Sistem hanya berfokus pada klasifikasi dan belum menyediakan visualisasi wilayah asal lagu
4	Penelitian ini	Klasifikasi asal daerah music tradisional Indonesia serta visualisasi wilayah	MFCC + Feature Fusion + XGBoost + GIS	-	Menggabungkan klasifikasi audio dan peta interaktif berbasis wilayah	-

2.2 Lagu Daerah

Lagu daerah merupakan karya musik yang berasal dari suatu daerah tertentu dan menjadi populer dinyanyikan baik oleh masyarakat daerah tersebut maupun masyarakat lainnya, dengan identitas pencipta yang pada umumnya sudah tidak diketahui lagi. Sebagai bagian dari kekayaan budaya, lagu daerah memainkan peran esensial sebagai representasi keindonesiaan yang tidak hanya berfungsi sebagai media hiburan, tetapi juga diterima sebagai artikulasi dari pluralitas serta kekayaan ras dan budaya bangsa. Dalam sejarahnya, lagu daerah telah digunakan sebagai instrumen untuk membentuk identitas budaya nasional karena lirik dan nadanya mengandung nilai-nilai yang mencerminkan karakter serta kearifan lokal masyarakat asalnya [4].

Keberagaman lagu daerah di Indonesia secara langsung dipengaruhi oleh posisi geografis negara yang sangat luas, yang memicu munculnya keragaman kelompok etnis dan bahasa daerah dengan jumlah mencapai ratusan jenis [14]. Kondisi ini menciptakan karakteristik musikal yang unik pada setiap wilayah, sebab perbedaan latar belakang budaya menghasilkan variasi dalam penggunaan instrumen tradisional serta struktur lagu yang spesifik. Setiap daerah mengembangkan ciri khas akustik tersendiri berdasarkan alat musik yang tersedia di lingkungan alam mereka, sehingga menghasilkan pola suara dengan variasi signifikan antara satu daerah dengan daerah lainnya [14]. Contoh lagu daerah di Indonesia tersebar di berbagai wilayah, seperti di Jawa Barat yang memiliki lagu Manuk Dadali dan Cing Cang Keling, serta di Sumatera Utara dengan lagu Sinanggar Tullo dan Butet yang mencerminkan kekayaan budaya daerah masing-masing [4].

2.2.1 Dinamika Nilai dalam lagu Daerah dari Masa ke Masa

Dinamika nilai dalam lagu daerah dari masa ke masa menunjukkan adanya perubahan dan perkembangan nilai-nilai budaya, sosial, dan moral yang terkandung dalam lagu tradisional seiring dengan perubahan kondisi masyarakat. Pada dasarnya, lagu daerah merupakan representasi kehidupan sosial budaya masyarakat yang mencerminkan nilai-nilai seperti kebersamaan, adat, moral, dan spiritual yang diwariskan secara turun-temurun [15]. Namun, seiring dengan perkembangan zaman, nilai-nilai tersebut mengalami pergeseran yang dipengaruhi oleh perubahan sosial dan budaya, seperti berkurangnya praktik tradisi lokal serta perubahan hubungan manusia dengan lingkungan [16]. Selain itu, kemajuan teknologi dan arus globalisasi turut memengaruhi eksistensi lagu daerah, yang menyebabkan perubahan dalam cara penyajian dan pemaknaannya di masyarakat modern [17]. Meskipun lagu daerah tetap memiliki peran penting sebagai media pelestarian budaya dan identitas

masyarakat, sehingga nilai-nilai yang terkandung di dalamnya perlu dijaga agar tidak hilang seiring perkembangan zaman[4].

2.2.2 Tantangan Lagu Daerah

Lagu daerah sebagai bagian dari warisan budaya menghadapi berbagai tantangan di era modern yang berkaitan dengan eksistensi dan daya tariknya di tengah perkembangan teknologi. Salah satu tantangan utama adalah menurunnya popularitas lagu daerah akibat dominasi musik populer global yang mudah diakses melalui media digital, sehingga menyebabkan pergeseran minat masyarakat, khususnya generasi muda, terhadap budaya lokal [17]. Selain itu, rendahnya keterlibatan generasi muda dalam mengenal dan melestarikan lagu daerah juga menjadi permasalahan, yang dipengaruhi oleh kurangnya penggunaan bahasa dan budaya lokal dalam kehidupan sehari-hari serta minimnya inovasi dalam penyampaian [18]. Tantangan lainnya terletak pada bentuk penyajian lagu daerah yang masih cenderung konvensional, sehingga kurang mampu bersaing dengan konten digital yang lebih menarik dan interaktif, padahal penggunaan media digital interaktif seperti multimedia atau teknologi berbasis visual terbukti dapat meningkatkan minat dan keterlibatan generasi muda terhadap musik tradisional [19]. Di sisi lain, keterbatasan digitalisasi dan pengemasan yang kurang relevan juga menyebabkan lagu daerah sulit berkembang dan beradaptasi dengan kebutuhan zaman. Hal ini menunjukkan bahwa tantangan utama lagu daerah di era modern tidak hanya terletak pada pelestariannya, tetapi juga pada bagaimana meningkatkan popularitas, keterlibatan generasi muda, serta penyajian yang lebih interaktif dan adaptif terhadap perkembangan teknologi.

2.3 Geographic Information System

Geographic Information System (GIS) didefinisikan sebagai teknologi komprehensif yang secara sistematis mampu menganalisis hubungan spasial dan dinamika temporal dari entitas dunia nyata melalui proses pengumpulan, penyimpanan, pemrosesan, hingga visualisasi informasi geografis [20]. Definisi ini sejalan dengan pandangan bahwa sistem informasi geografis merupakan sistem berbasis komputer yang digunakan untuk mengolah, memanipulasi, menganalisis, dan menyajikan data geografis yang berkaitan dengan lokasi di permukaan bumi [21]. Dengan kemampuan tersebut, GIS dirancang untuk mendukung proses pengambilan keputusan, khususnya dalam perencanaan dan pengelolaan yang efektif, efisien, serta berkelanjutan di berbagai sektor [22].

Seiring dengan perkembangan teknologi, pemanfaatan GIS semakin meluas dan memberikan kontribusi signifikan dalam berbagai bidang [20]. Integrasi GIS dengan

teknologi berbasis web (WebGIS) turut meningkatkan aksesibilitas, kolaborasi, dan efisiensi dalam analisis data geospasial, sehingga mampu mendukung proses pembangunan berkelanjutan secara lebih optimal [23]. Salah satu fungsi GIS yang paling banyak digunakan adalah kemampuannya dalam memvisualisasikan data geografis secara interaktif, sehingga informasi spasial dapat dipahami dengan lebih mudah oleh pengguna.

2.4 Music Information Retrieval (MIR)

Music Information Retrieval (MIR) merupakan bidang penelitian multidisiplin yang berfokus pada proses pengolahan, analisis, dan pengambilan informasi dari data musik menggunakan teknik komputasi. MIR menggabungkan berbagai disiplin ilmu seperti pemrosesan sinyal digital, kecerdasan buatan, pembelajaran mesin, serta teori musik untuk mengekstraksi informasi bermakna dari sinyal audio maupun metadata musik. Melalui pendekatan ini, sistem komputer dapat mengidentifikasi karakteristik musik seperti genre, tempo, instrumen, emosi hingga stuktur musikal secara otomatis. Dengan meningkatnya jumlah koleksi musik digital di berbagai platform, MIR menjadi teknologi penting dalam membatu pengelolaan dan pencarian informasi musik secara efisien dan terstruktur [24], [25].

Pengembangan MIR didorong oleh kebutuhan untuk mengelola dan menganalisis data musik digital yang jumlahnya semakin besar [26]. Berbagai industri layanan streaming musik, pengembang aplikasi musik, serta peneliti di bidang kecelenerrdasan buatan memanfaatkan teknologi ini untuk melakukan klasifikasi audio secara otomatis. MIR juga banyak digunakan dalam penelitian akademik untuk memahami pola dalam data dan mengembangakan sistem pengenalan musik berbasis *machine learning* [25]. Selain itu, MIR memiliki keunggulan dalam kemampuannya untuk mengintegrasikan berbagai jenis data musik, baik dalam bentuk sinyal audio maupun metadata, sehingga memungkinkan pemahaman yang lebih komprehensif terhadap suatu karya musik. Pendekatan ini menjadikan MIR fleksibel untuk digunakan dalam berbagai konteks, mulai dari pengelolaan arsip musik digital hingga eksplorasi informasi musikal secara luas. Dengan dukungan teknik komputasi modern, MIR juga mampu menangani data dalam skala besar secara efisien, sehingga sangat relevan dengan perkembangan ekosistem musik digital saat ini [25].

Di sisi lain MIR masih menghadapi sejumlah keterbatasan yang menjadi tantangan dalam penerapannya. Salah satu kekurangan utama terletak pada kompleksitas karakteristik musik yang bersifat subjektif dan dipengaruhi oleh latar belakang budaya, sehingga persepsi terhadap elemen musik seperti melodi, harmoni, dan emosi tidak selalu dapat

direpresentasikan secara universal oleh sistem komputasi [25]. Selain itu, keterbatasan dataset yang tersedia, yang umumnya didominasi oleh jenis musik tertentu seperti musik Barat, menyebabkan kurangnya representasi terhadap keberagaman musik global dan berpotensi menimbulkan bias pada sistem yang dikembangkan [25]. Di sisi lain, variasi kualitas data, perbedaan format audio, serta ketidakkonsistenan metadata juga menjadi kendala dalam menghasilkan representasi informasi yang akurat, ditambah dengan belum adanya standar evaluasi yang sepenuhnya seragam sehingga menyulitkan perbandingan kinerja antar sistem MIR secara objektif[26].

2.5 Digital Signal Processing

Digital Signal Processing (DSP) adalah bidang ilmu yang mempelajari teknik pengolahan sinyal dalam bentuk digital untuk mengekstraksi informasi yang terkandung di dalamnya. Dalam konteks audio, sinyal suara yang awalnya berupa gelombang analog terlebih dahulu dikonversi menjadi sinyal digital melalui proses sampling agar dapat diproses menggunakan komputer. Proses ini memungkinkan sinyal suara dianalisis secara matematis untuk memperoleh karakteristik tertentu yang dapat digunakan dalam berbagai aplikasi seperti pengenalan suara, klasifikasi musik, dan identifikasi pola akustik [27].

Proses ini dimulai dengan tahap digitalisasi melalui *sampling* dan *kuantisasi*. Agar sinyal dapat direpresentasikan secara akurat tanpa terjadi fenomena *aliasing*, frekuensi pengambilan sampel (*sampling rate* atau f_s) harus memenuhi kriteria *Nyquist-Shannon*, yaitu setidaknya dua kali lipat dari frekuensi tertinggi dalam sinyal ($f_s \geq 2f_{max}$) [28]. Sinyal audio mentah yang ditangkap melalui mikrofon umumnya berada dalam domain waktu (*time-domain*), di mana informasi yang tersaji hanyalah fluktuasi amplitudo terhadap waktu. Namun, karakteristik unik musik tradisional seperti harmonisasi instrumen gamelan atau teknik vokal tertentu lebih mudah diidentifikasi melalui kandungan frekuensinya (*frequency-domain*). Transisi antar domain ini dicapai menggunakan algoritme *Fast Fourier Transform* (FFT), yang membedah sinyal kompleks menjadi komponen-komponen frekuensi penyusunnya [29]. Karena sinyal musik bersifat non-stasioner, di mana kandungan frekuensinya berubah seiring berjalannya waktu, maka diterapkan teknik *Short-Time Fourier Transform* (STFT). Pada teknik ini, sinyal audio dibagi menjadi segmen-segmen kecil atau bingkai (*frames*) yang tumpang tindih (*overlap*) untuk dianalisis secara terpisah, sehingga perubahan karakter suara dari detik ke detik dapat tertangkap dengan presisi [30].

Tujuan akhir dari DSP dalam penelitian ini adalah mereduksi kompleksitas data audio asli menjadi sekumpulan fitur representatif. Proses ini menghilangkan informasi yang

tidak relevan (seperti *noise*) dan menonjolkan karakteristik unik dari alat musik atau gaya vokal tradisional Indonesia. Fitur-fitur hasil olahan DSP inilah yang nantinya akan digunakan sebagai input bagi model XGBoost untuk menentukan asal daerah lagu tersebut [31].

2.6 Segmentasi

Segmentasi audio merupakan tahapan pra-pemrosesan (*pre-processing*) krusial dalam pengolahan sinyal audio yang bertujuan untuk membagi aliran audio (*audio stream*) menjadi bagian-bagian kecil (*frame*) atau segmen yang homogen berdasarkan karakteristik suaranya. Proses ini dilakukan sebelum tahap ekstraksi fitur guna memisahkan berbagai jenis elemen suara, seperti ucapan (*speech*), musik, suara lingkungan (*environmental sound*), hingga bagian hening (*silence*), agar dapat dianalisis lebih lanjut secara spesifik [32]elain itu, proses segmentasi dapat dilakukan melalui *silence removal*, yaitu dengan menghilangkan bagian diam (*silent areas*) dari rekaman audio sehingga diperoleh segmen yang merepresentasikan kejadian audio (*audio events*)[33]

2.7 Ekstraksi Fitur

Ekstraksi fitur merupakan proses penting dalam pengolahan sinyal audio yang bertujuan untuk mendukung analisis dan klasifikasi data audio. Proses ini mengubah sinyal mentah menjadi representasi yang lebih ringkas namun tetap mempertahankan informasi utama sehingga memudahkan proses pengolahan lebih lanjut. Selain itu, ekstraksi fitur juga berfungsi untuk menyederhanakan data dengan mengambil karakteristik penting dari sinyal agar lebih efisien untuk diproses dalam sistem *machine learning*[34]. Ekstraksi fitur merupakan tahap fundamental dalam pemrosesan audio karena sangat mempengaruhi kualitas representasi data yang digunakan dalam model[35].

2.7.1 Time Domain

Dalam pengolahan sinyal audio, fitur ekstraksi dapat dibagi menjadi beberapa kategori, salah satunya adalah domain waktu (*time domain*). Fitur domain waktu merupakan fitur yang diperoleh langsung dari sinyal audio dalam bentuk aslinya terhadap waktu tanpa melalui transformasi ke domain lain. Pendekatan ini digunakan untuk menangkap karakteristik dasar sinyal seperti energi dan perubahan amplitudo dalam domain waktu. Dibandingkan dengan metode lain, analisis domain waktu cenderung lebih sederhana karena dilakukan langsung pada sinyal asli[34]. Berikut ini adalah beberapa domain waktu yang umum digunakan[34]:

1. Energy

Energy (energi) merupakan fitur dasar dalam analisis audio yang mengukur kekuatan atau intensitas sinyal dalam suatu durasi waktu tertentu (*frame*). Nilai energi menunjukkan besarnya amplitudo sinyal; semakin besar amplitudonya, maka semakin tinggi energi yang dihasilkan. Fitur ini sangat berguna untuk membedakan segmen suara yang aktif dengan segmen diam (*silence*). Nilai energi diperoleh dari jumlah kuadrat amplitudo sampel dalam satu *frame* sesuai persamaan berikut:

$$E(i) = \sum_{n=1}^{W_L} |x_i(n)|^2 \quad \text{.....(1)}$$

Keterangan:

$E(i)$: Energi (atau *power*) pada *frame* ke-i

W_L: Panjang *frame* (jumlah sampel dalam satu bingkai).

$x_i(n)$: Amplitudo sampel ke-n pada *frame* ke-i

2. Energy Entropy

Energy Entropy adalah fitur yang digunakan untuk mengukur tingkat penyebaran atau distribusi energi dalam sebuah *frame* audio. Fitur ini menunjukkan apakah energi tersebar secara merata di seluruh durasi *frame* atau terkonsentrasi hanya pada bagian tertentu saja. Untuk menghitungnya, sebuah *frame* dibagi menjadi *K* bagian kecil (*sub-frame*). Proses perhitungannya mengikuti tiga tahapan rumus berikut:

a. Menghitung Total Energi satu *frame* dengan menjumlahkan energi dari seluruh *sub-frame*:

$$E_{shortFrame_i} = \sum_{k=1}^{K} E_{subFame_k} \quad \text{.....(2)}$$

b. Melakukan Normalisasi untuk mendapatkan nilai probabilitas energi tiap *sub-frame* (*ej*):

$$e_j = \frac{E_{subFrame_j}}{E_{shortFrame_i}} \quad \text{.....(3)}$$

c. Menghitung Nilai Entropy (*H*) menggunakan persamaan Shannon Entropy sebagai hasil akhir:

$$H(i) = -\sum_{j=1}^{K} e_j \cdot \log_2(e_j) \quad \text{.....(4)}$$

Keterangan:

$H(i)$: Nilai *Energy Entropy* pada *frame* ke-i

$E_{shortFrame_i}$: Total energi pada *frame* singkat ke-i

$E_{shortFrame_i}$: Energi pada *sub-frame* ke-j

e_j: Nilai probabilitas energi (hasil normalisasi) pada *sub-frame* ke- j

K: Jumlah total *sub-frame* dalam satu *frame*.

3. Zero Crossing Rate (ZCR)

Zero Crossing Rate (ZCR) merupakan jumlah perubahan tanda sinyal dari positif ke negatif atau sebaliknya dalam suatu frame yang dinormalisasi terhadap panjang frame. Penghitungan ZCR didefinisikan melalui persamaan berikut:

$$Z(i) = \frac{1}{2W_L}\sum_{n=1}^{W_L}|\text{sgn}[x_i(n)] - \text{sgn}[x_i(n-1)]| \quad (5)$$

Keterangan:

$z_{(i)}$: Nilai ZCR pada *frame* ke-i

$w_{(L)}$: Panjang atau jumlah sampel dalam satu *frame*

$x_i(n)$: Nilai sampel ke-n pada *frame* ke-i

sgn: Fungsi untuk menentukan tanda (+ atau -) dari amplitudo sampel.

2.7.2 Frequency Domain

Selain domain waktu, pendekatan lain yang umum digunakan adalah domain frekuensi (*frequency domain*). Fitur domain frekuensi diperoleh dengan mengubah sinyal dari domain waktu ke domain frekuensi menggunakan transformasi seperti *Fast Fourier Transform* (FFT) untuk menganalisis distribusi spektrum sinyal [34]. Representasi ini memungkinkan sinyal dianalisis berdasarkan karakteristik frekuensinya sehingga memberikan informasi yang lebih rinci mengenai komponen spektral. Pendekatan ini banyak digunakan dalam pemrosesan audio karena mampu merepresentasikan karakteristik spektral sinyal, meskipun memiliki kompleksitas komputasi yang lebih tinggi dibandingkan metode domain waktu[35]. Terdapat beberapa fitur dalam domain frekuensi yang umum digunakan untuk merepresentasikan karakteristik sinyal, antara lain spectral centroid, spectral spread, spectral entropy, spectral flux, dan spectral rolloff[34]. Fitur-fitur tersebut dijelaskan sebagai berikut:

1. Spectral Centroid

Spectral centroid adalah fitur yang menunjukkan pusat massa dari spektrum magnitudo suatu sinyal audio. Fitur ini digunakan untuk mengukur karakteristik bentuk spektrum, di mana nilai centroid yang lebih tinggi menunjukkan tekstur suara yang lebih "cerah" (*bright*) karena

didominasi oleh frekuensi tinggi. Sebaliknya, nilai yang rendah menunjukkan suara yang lebih berat atau "gelap" (*dark*)[36].

Nilai *Spectral Centroid* pada sebuah *frame* dihitung menggunakan persamaan berikut:

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] \cdot n}{\sum_{n=1}^{N} M_t[n]} \quad (6)$$

Keterangan:

C_t: Nilai Spectral Centroid pada frame ke-t.

$M_t[n]$: Nilai magnitudo dari *Fourier Transform* pada *frame t* dan indeks frekuensi (*frequency bin*) *n*.

n: Indeks *frequency bin*.

N: Jumlah total *frequency bin*.

2. Spectral Spread

Spectral spread mengukur penyebaran frekuensi di sekitar spectral centroid, sehingga menunjukkan seberapa luas distribusi energi dalam spektrum. Nilai spread yang besar menunjukkan variasi frekuensi yang lebih lebar, sedangkan nilai kecil menunjukkan energi terkonsentrasi pada frekuensi tertentu[36].

$$S_i = \sqrt{\frac{\sum_{k=1}^{Wf_L} (k - C_i)^2 X_i(k)}{\sum_{k=1}^{Wf_L} X_i(k)}} \quad (7)$$

Keterangan:

S_i: Nilai *Spectral Spread* untuk *frame* ke-i.

C_i: *Spectral Centroid* (pusat massa spektrum). Ini adalah titik acuan penyebaran.

k: Indeks frekuensi (*frequency bin*).

$X_i(k)$:Magnitudo atau energi pada frekuensi k

$(k - C_i)^2$: Selisih kuadrat antara frekuensi saat ini dengan pusatnya. Inilah yang menentukan "lebar" penyebaran.

3. Spectral Entropy

Spectral entropy merupakan ukuran ketidakteraturan distribusi energi dalam spektrum frekuensi yang dihitung berdasarkan konsep entropy. Nilai entropy yang tinggi menunjukkan distribusi energi yang merata (noise-like), sedangkan nilai rendah menunjukkan sinyal yang lebih terstruktur [37].

Nilai *Spectral Entropy* dihitung menggunakan persamaan berikut:

$$H(X) = -\sum_{i=1}^{N} P(f_i) \log_2\big(P(f_i)\big) \quad (8)$$

Keterangan:

$H(X)$: Nilai *Spectral Entropy* dari sinyal *X*.

$P(f_i)$: Nilai energi yang telah dinormalisasi pada *frequency bin* ke-*i.*

$\log_2$: Logaritma basis 2.

4. Spectral Flux

   Spectral flux mengukur perubahan spektrum frekuensi antar frame berturut-turut, sehingga dapat digunakan untuk mendeteksi perubahan dinamika atau transisi dalam sinyal audio [37].

   Persamaan matematis untuk *Spectral Flux* adalah sebagai berikut:

   $F_t = \sum_{n=1}^{N}(N_t[n] - N_{t-1}[n])^2$ .....(9)

   Keterangan:

   F_t: Nilai *Spectral Flux* pada *frame* ke-*t*.

   $N_t[n]$: Magnitudo spektrum yang dinormalisasi pada *frame* ke-*t* dan indeks frekuensi *n*.

   $N_{t-1}[n]$: Magnitudo spektrum yang dinormalisasi pada *frame* sebelumnya (*t*-1) dan indeks frekuensi *n*.

5. Spectral Rolloff

   Spectral rolloff merupakan frekuensi batas di mana sebagian besar energi spektrum berada (biasanya 85% dari total energi). Fitur ini digunakan untuk membedakan jenis sinyal seperti speech dan music berdasarkan distribusi energinya [37].

   $\sum_{n=1}^{R_t} M_t[n] = 0{,}85 \cdot \sum_{n=1}^{N} M_t[n]$.....(10)

   Keterangan:

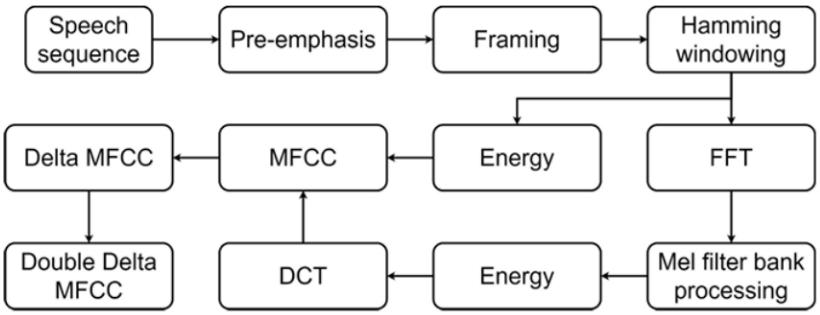
   R_t: Nilai Spectral Rolloff pada frame ke-*t*.

   0,85: Konstanta ambang batas persentase energi (dalam hal ini 85%).

2.7.3 MFCC

Mel-Frequency Cepstral Coefficient (MFCC) merupakan salah satu metode ekstraksi ciri yang banyak digunakan dalam pengolahan sinyal suara untuk merepresentasikan karakteristik penting dari sinyal audio dalam bentuk parameter numerik. MFCC bekerja dengan mengekstraksi nilai cepstral dari sinyal suara sehingga dapat digunakan sebagai fitur dalam sistem pengenalan suara [38]. Secara umum, MFCC digunakan karena kemampuannya dalam meniru sistem pendengaran manusia melalui penggunaan skala Mel, yaitu skala frekuensi no-linear yang lebih sensitif terhadap frekuensi rendah dibandingkan frekuensi tinggi [39]. Hal ini menunjukkan MFCC sangat efektif dalam berbagai aplikasi seperti pengenalan ucapan, identifikasi pembicara, dan klasifikasi audio [40].

Terdapat beberapa tahapan yang perlu dilakukan dalam proses MFCC sehingga didapatkan nilai ekstraksi ciri. Alur proses MFCC ditunjukkan pada Gambar 2.1 berikut.



Gambar 2. 1 Proses ekstraksi fitur menggunakan metode MFCC[40].

Tahapan ini merupakan proses standar dalam ekstraksi fitur MFCC yang digunakan untuk memperoleh representasi sinyal yang lebih kompak dan informatif [39].

1. Pre-emphasis

   Pre-emphasis merupakan tahap awal dalam ekstraksi MFCC yang bertujuan untuk meningkatkan energi pada frekuensi tinggi dalam sinyal suara. Hal ini diperlukan karena secara alami sinyal suara memiliki energi yang lebih dominan pada frekuensi rendah [39]

   Secara matematis, pre-emphasis dinyatakan sebagai:

   $y[n] = x[n] - \alpha x[n-1]$ .....(11)

   Keterangan:

   $x[n]$= sinyal input

   $y[n]$= sinyal hasil pre-emphasis

   α= konstanta pre-emphasis (biasanya 0.9–1.0)

   Tahapan ini berfungsi untuk memperbaiki rasio sinyal terhadap noise dan menyeimbangkan spektrum frekuensi sebelum diproses lebih lanjut [38].

2. Framing

   Framing adalah proses pembagian sinyal suara menjadi segmen-segmen kecil (frame) dalam interval waktu tertentu. Hal ini dilakukan karena sinyal suara bersifat non-stasioner, namun dapat dianggap stasioner dalam jangka waktu yang sangat pendek [39]. Umumnya, panjang frame berkisar antara 20–40 ms dengan overlap antar frame untuk menjaga kontinuitas informasi [40]. Dengan *framing*, analisis spektral dapat dilakukan secara lebih akurat karena setiap frame merepresentasikan kondisi sinyal yang relatif stabil.

3. Windowing

   Setelah proses *framing*, setiap frame dikalikan dengan fungsi *window* untuk mengurangi diskontinuitas pada batas frame. Salah satu fungsi *window* yang umum digunakan adalah *Hamming window* [38].

   Rumus Hamming window:

$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$ .....(12)

Keterangan:

$w[n]$: nilai window,

n: 0,1,2…., M-1,

N: panjang frame.

Penggunaan window ini bertujuan untuk mengurangi efek *spectral leakage*, yaitu penyebaran energi spektrum akibat pemotongan sinyal secara tiba-tiba [39].

4. Fast Fourier Transform (FFT)

FFT merupakan proses transformasi sinyal dari domain waktu ke domain frekuensi. Tahapan ini menghasilkan spektrum frekuensi yang menunjukkan distribusi energi sinyal pada berbagai frekuensi [40].

Secara matematis:

$X(k) = \sum_{n=0}^{N-1} x_w[n] \cdot e^{-j2\pi kn/N}$ .....(13)

Keterangan:

$X[k]$: *Fast Fourier Transform*

$x_w[n]$: sampel sinyal hasil windowing

n: indeks waktu dalam domain waktu

J: bilangan imajiner

N: jumlah sampel dalam satu frame

FFT sangat penting dalam MFCC karena memungkinkan analisis komponen frekuensi yang menjadi dasar pemrosesan pada tahap berikutnya [41].

5. Mel Filter Bank

Mel Filter Bank merupakan tahap yang mengubah spektrum frekuensi menjadi skala *Mel* yang sesuai dengan persepsi pendengaran manusia [39].

Hubungan antara frekuensi linear dan skala Mel dinyatakan sebagai:

$F_{mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$ .....(14)

Keterangan:

F_{mel}: Koefisien filter bank,

f= frekuensi

Filter bank terdiri dari sejumlah filter berbentuk segitiga yang lebih rapat pada frekuensi rendah dan lebih renggang pada frekuensi tinggi, sesuai dengan karakteristik sistem pendengaran manusia [40].

Output dari tahap ini adalah energi pada masing-masing filter yang kemudian dikonversi ke bentuk logaritmik.

6. Discrete Cosine Transform (DCT)

   DCT merupakan langkah terakhir ekstraksi fitur dengan MFCC, digunakan untuk mengubah log energi hasil *Mel* Filter Bank ke dalam domain *cepstral*. Transformasi ini bertujuan untuk menghasilkan koefisien yang tidak saling berkorelasi dan lebih efisien untuk digunakan sebagai fitur [41].

   Rumus DCT:

   $C_n = \sum_{k=1}^{K} \log(S_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right]$ .....(15)

   Keterangan:

   C_n= koefisien MFCC

   S_k= energi filter ke-k

   K= jumlah total *Mel filter Bank*

   Koefisien MFCC yang dihasilkan biasanya diambil dalam jumlah tertentu (misalnya 12–25 koefisien) karena sudah mampu merepresentasikan informasi penting dari sinyal suara [41].

2.8 Feature Fusion

Feature fusion merupakan salah satu konsep inti dalam *machine learning* yang berkaitan dengan penggabungan informasi dari berbagai sumber atau modalitas data. *Feature fusion* adalah proses mengambil fitur-fitur yang diekstraksi dari suatu data dan menggabungkannya menjadi satu representasi fitur yang lebih diskriminatif dibandingkan fitur-fitur inputnya secara terpisah. Dalam konteks yang lebih luas, data *fusion* merupakan penggabungan data dari modalitas-modalitas berbeda yang menyediakan sudut pandang terpisah terhadap suatu fenomena yang sama untuk menyelesaikan sebuah masalah inferensi, dengan tujuan mencapai tingkat kesalahan yang lebih rendah dibandingkan pendekatan *unimodal* [42].

Fungsi utama *feature fusion* dalam *pipeline machine learning* adalah memperkaya representasi data yang digunakan model untuk belajar dan membuat prediksi. Strategi data *fusion* merupakan konsep sentral dalam *multimodal learning* yang dapat diterapkan pada berbagai tugas *machine learning* dengan berbagai jenis modalitas [43]. Modalitas-modalitas tersebut dapat saling melengkapi, sehingga pemilihan strategi data *fusion* yang tepat dapat meningkatkan performa model secara signifikan. Bahkan, terdapat permasalahan *machine learning* yang tidak dapat diselesaikan

tanpa mempertimbangkan seluruh modalitas secara bersama-sama [43]. Dalam literatur *machine learning*, terdapat tiga jenis utama strategi *feature fusion* berdasarkan tahap penggabungannya, yaitu *late fusion, early fusion*, dan *sketch*. Pada teknik *late fusion*, setiap modalitas dipelajari secara independen dan digabungkan menjelang tahap pengambilan keputusan. Pada teknik *early fusion*, seluruh modalitas digabungkan terlebih dahulu sebelum proses pelatihan model. Sementara itu, pada teknik *sketch*, modalitas ditransformasikan ke dalam ruang bersama (*common space*), bukan sekadar digabungkan secara langsung [43].

2.9 Machine Learning

Machine Learning merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pengembangan metode dan algoritma yang memungkinkan sistem komputer belajar dari data tanpa diprogram secara eksplisit. Melalui proses pembelajaran dari data tersebut, sistem dapat mengenali pola dan menghasilkan prediksi atau keputusan secara otomatis. Dalam *machine learning*, model dibangun dengan memanfaatkan data historis untuk mempelajari hubungan antara variabel *input* dan *output* sehingga model dapat digunakan untuk memprediksi data baru yang belum pernah dilihat sebelumnya. Pendekatan ini banyak digunakan dalam berbagai bidang seperti pengolahan citra, pengenalan suara, sistem rekomendasi, dan klasifikasi data[44]. *Machine learning* memungkinkan komputer untuk mempelajari pola dari data melalui proses pelatihan (*training*) sehingga model yang dihasilkan dapat melakukan generalisasi terhadap data baru. Dengan kata lain, model *machine learning* tidak hanya mengingat data pelatihan, tetapi juga mampu melakukan prediksi terhadap data yang belum pernah ditemui sebelumnya [45].Secara umum, machine learning dapat dibagi menjadi beberapa jenis berdasarkan cara sistem mempelajari data, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

Supervised learning merupakan metode machine learning yang menggunakan dataset yang telah memiliki label atau target. Pada metode ini, model dilatih menggunakan pasangan data *input* dan *output* sehingga model dapat mempelajari hubungan antara keduanya. Setelah proses pelatihan selesai, model dapat digunakan untuk memprediksi output dari data baru. *Supervised learning* biasanya digunakan untuk dua jenis permasalahan utama, yaitu klasifikasi dan regresi. Klasifikasi digunakan ketika output berupa kategori tertentu, sedangkan regresi digunakan ketika output berupa nilai kontinu[46].

Unsupervised learning merupakan metode machine learning yang menggunakan data tanpa label. Tujuan utama metode ini adalah untuk menemukan pola, struktur, atau hubungan tersembunyi dalam data. Metode ini sering digunakan untuk proses *clustering* atau pengelompokan data berdasarkan kemiripan karakteristik. Selain itu, *unsupervised learning* juga digunakan dalam teknik reduksi dimensi dan analisis pola data[47].

Reinforcement learning merupakan metode pembelajaran yang dilakukan melalui proses interaksi antara agen dan lingkungan. Dalam metode ini, sistem belajar melalui mekanisme *trial* and *error* dengan mendapatkan *reward* atau *penalty* berdasarkan tindakan yang dilakukan. Tujuan dari *reinforcement learning* adalah untuk menemukan strategi atau kebijakan terbaik yang dapat memaksimalkan nilai *reward* dalam jangka panjang. Metode ini sering digunakan dalam bidang robotika, permainan komputer, dan sistem pengambilan keputusan otomatis.[47]

Berdasarkan jenis pembelajarannya, penelitian ini menerapkan pendekatan *supervised learning*. Dalam pendekatan ini, model dilatih menggunakan dataset yang telah memiliki label target yang jelas (seperti nama daerah asal lagu). Algoritme akan berusaha meminimalkan selisih antara prediksi model dengan label asli melalui optimasi fungsi kerugian (*loss function*), sehingga model mampu memetakan hubungan antara fitur akustik dengan identitas daerahnya secara akurat [46]. Selain itu, dalam *supervised learning*, proses pembelajaran tidak hanya berhenti pada pembentukan model dari data berlabel, tetapi juga melihat tahapan optimasi yang kompleks untuk menghasilkan model yang memiliki kemampuan generalisasi yang baik. Secara formal, tujuan utama dari *supervised learning* adalah membangun suatu fungsi pemetaan antara data input dan output sehingga model mampu melakukan prediksi secara akurat terhadap data baru yang belum pernah dilihat sebelumnya. Proses ini umumnya dilakukan dengan membagi dataset menjadi data pelatihan (*training data*) dan data pengujian (*testing data*), dimana data pelatihan digunakan untuk membangun model, sedangkan data pengujian digunakan untuk evaluasi kinerja model terhadap data yang tidak dikenal [46].

2.10 Klasifikasi

Klasifikasi merupakan salah satu teknik dalam *data mining* dan pembelajaran mesin yang bertujuan untuk mengelompokkan data ke dalam kategori atau kelas tertentu berdasarkan karakteristik yang dimilikinya. Proses ini dilakukan dengan mempelajari pola dari data yang telah memiliki label kelas agar sistem mampu memprediksi kategori dari data baru yang belum diketahui sebelumnya. Dalam analisis data modern, teknik ini banyak diterapkan untuk memecahkan berbagai permasalahan kompleks seperti pengenalan gambar, analisis teks, hingga pengolahan sinyal audio [48]. Melalui pendekatan ini, klasifikasi menjadi komponen utama dalam sistem pengenalan pola dan proses pengambilan keputusan berbasis data secara otomatis.

Secara teknis, sistem klasifikasi bekerja dengan memanfaatkan tiga komponen utama, yaitu data, fitur, dan label kelas. Data merupakan kumpulan sampel yang digunakan dalam proses pembelajaran, sedangkan fitur adalah atribut atau karakteristik yang merepresentasikan informasi penting dari data tersebut. Label kelas menunjukkan kategori yang menjadi tujuan akhir dari proses identifikasi. Dalam pendekatan *supervised learning*, proses klasifikasi dilakukan dengan memanfaatkan *dataset* yang telah memiliki pasangan fitur dan label sehingga model dapat

mempelajari hubungan antara keduanya untuk menghasilkan prediksi yang akurat pada data baru[49].

Permasalahan klasifikasi dapat dibedakan berdasarkan jumlah kategori yang terlibat di dalamnya. Klasifikasi biner (*binary classification*) merupakan jenis klasifikasi yang hanya memiliki dua kemungkinan kelas, seperti kategori positif dan negatif. Sementara itu, klasifikasi multikelas (*multiclass classification*) melibatkan lebih dari dua kategori sehingga setiap data dapat diklasifikasikan ke dalam salah satu dari beberapa kelas yang tersedia [49]. Jenis klasifikasi multikelas ini sangat relevan dalam menangani permasalahan pengelompokan data dengan kategori yang kompleks, seperti penentuan asal daerah dari berbagai provinsi yang berbeda.

Secara umum, sistem klasifikasi terdiri dari beberapa tahapan utama yang meliputi pengumpulan data, proses pelatihan (*training*), dan proses prediksi (*classification*). Pada tahap awal, dataset yang telah berlabel digunakan sebagai data pelatihan untuk membangun model klasifikasi. Model tersebut kemudian mempelajari hubungan antara fitur dan kelas untuk mengenali pola-pola tertentu. Setelah proses pelatihan selesai, model dapat digunakan untuk memprediksi kelas dari data baru berdasarkan fitur yang dimiliki [49]. Tahapan ini menentukan kemampuan model dalam mengidentifikasi kategori suatu data secara otomatis dan konsisten.

2.11 XGBoost

Extreme Gradient Boosting (XGBoost) merupakan algoritma *machine learning* berbasis *ensemble learning* yang dikembangkan dari metode *Gradient Boosting Decision Tree* (GBDT) [50]. *Ensemble learning* sendiri adalah pendekatan yang menggabungkan beberapa model untuk meningkatkat performa prediksi dibandingkan model tunggal [51]. XGBoost bekerja dengan membangun model secara bertahap, dimana setiap model baru berfungsi untuk memperbaiki kesalahan dari model sebelumnya melalui proses *boosting* [50]. Pendekatan ini memungkinkan model untuk belajar dari error sebelumnya sehingga menghasilkan prediksi yang lebih akurat.

Secara umum, model XGBoost dapat dinyatakan sebagai penjumlahan dari beberapa fungsi pohon keputusan

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad \text{.....(16)}$$

Keterangan:

$\hat{y}_i$: Prediksi akhir ke-i.

K: Jumlah total pohon (*trees*).

f_k: Fungsi pohon keputusan ke-k.

x_i: Input data atau fitur yang digunakan untuk prediksi

Dimana f_k merupakan decision tree ke-k dan K adalah jumlah total tree dalam model [50].

Konsep ini menunjukan bahwa XGBoost tidak bergantung pada satu model saja, melainkan kombinasi banyak model sederhana yang bekerja secara kolektif untuk menghasilkan prediksi yang optimal.

2.11.1 Prinsip Kerja XGBoost

Prinsip kerja XGBoost didasarkan pada teknik *gradient boosting*, yaitu metode optimasi yang menimbulkan fungsi loss secara interatif menggunakan gradien [52].

Pada setiap iterasi ke-t, model diperbaharui dengan menambahkan fungsi baru:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(xi) \quad (17)$$

Keterangan:

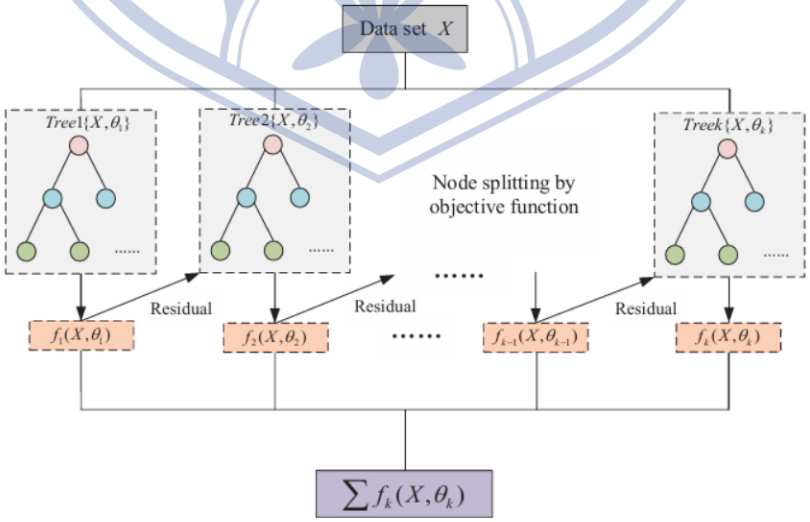
$\hat{y}_i^{(t)}$: prediksi pada iterasi ke-*t*.

$\hat{y}_i^{(t-1)}$: Prediksi dari iterasi sebelumnya.

$f_t(xi)$: pohon baru yang ditambahkan untuk memperbaiki error.

Persamaan ini menunjukkan bahwa setiap model baru $f_t(x_i)$ bertugas untuk memperbaiki kesalahan dari model sebelumnya. Dengan kata lain, XGBoost tidak membangun model dari nol pada setiap iterasi, tetapi melakukan perbaikan secara bertahap.

Keunggulan utama XGBoost dibandingkan metode boosting tradisional adalah penggunaan turunan order kedua (*second-order gradient*) dalam proses optimasi. dengan mempertimbangkan perbandingan gradien pertama dan kedua, model dapat memahami arah dan kecepatan perubahan error sehingga proses pembelajaran menjadi lebih cepat dan stabil [50], [52]). Secara visual, tahapan proses perbaikan model secara bertahap ini diilustrasikan pada Gambar 2.2 berikut.



Gambar 2. 2 Flowchart XGBoost[53]

Alur proses yang terjadi di dalam flowchart dapat dijelaskan sebagai berikut[53]:

1) Inisialisasi Data: Alur dimulai dengan memasukkan *Dataset X* kedalam sistem untuk diproses oleh serangkaian pohon keputusan (*decision trees*).
2) Pembangunan Pohon Secara Aditif: XGBoost membangun model secara bertahap dengan menambahkan satu pohon baru pada setiap iterasi. Gambar tersebut menunjukkan transisi dari $Tree_1$ hingga $Tree_k$.
3) Pembelajaran Residual: Setiap pohon baru bertugas untuk mempelajari dan memperbaiki kesalahan atau *residual* dari prediksi pohon sebelumnya. Hal ini terlihat pada panah Residual yang menghubungkan satu tahap ke tahap berikutnya dalam flowchart.
4) Optimasi *Node Splitting*: Pemisahan cabang (*node splitting*) di dalam setiap pohon dipandu oleh fungsi objektif yang menghitung keuntungan (*gain*) maksimal untuk meminimalkan *loss*.
5) Agregasi Skor Akhir: Setelah K jumlah pohon terbentuk, fitur dari sampel prediksi akan melewati setiap pohon untuk mendapatkan skor pada setiap *leaf node*.
6) Prediksi *Final*: Langkah terakhir adalah menjumlahkan seluruh skor dari setiap pohon $\sum_{k=1}^{K} f_k(X, \theta_k)$ untuk menghasilkan nilai prediksi akhir yang optimal.

2.11.2 Fungsi Objektif dan Regularisasi

XGBoost menggunakan fungsi objektif yang terdiri dari dua komponen utama, yaitu fungsi *loss* dan fungsi regularisasi:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (18)$$

Kereangan:

$\mathcal{L}(\phi)$: Fungsi objektif total.

$l(y_i, \hat{y}_i)$: Fungsi loss (perbedaan nilai aktual $y_{i,}$ dan prediksi $\hat{y}_i$).

$\Omega(f_k)$: Istilah regularisasi (penalty kompleksitas pohon)

Persamaan tersebut menunjukkan bahwa tujuan utama model adalah meminimalkan kesalahan prediksi sekaligus mengontrol kompleksitas model. Komponen pertama mengukur seberapa jauh hasil prediksi dari nilai sebenarnya, sedangkan komponen kedia berfungsi sebagai penalti terhadap kompleksitas model [52].

Fungsi regularisasi pada XGBoost didefenisikan sebagai:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (19)$$

Keterangan:

T: jumlah daun pada pohon (number of leaves).

w_j : Bobot (*score*) pada setiap daun.

γ, λ: Parameter kontrol untuk mencegah *overfitting*.

Secara intuitif, persamaan ini menunjukkan bahwa model akan ”dihukum” jika terlalu kompleks. Hal ini penting karena model yang terlalu kompleks cenderung hanya cocok pada training dan tidak mampu melakukan generalisasi dengan baik pada data baru.

2.11.3 Optimasi dengan Second-Order Gradient

Untuk mengoptimalkan fungsi objektif, XGBoost menggunakan pendekatan *second-order Taylor expansion*:

$$\widetilde{\mathcal{L}^{(t)}} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (20)$$

Keterangan:

g_i: Gradien (turunan pertama).

h_i: Hessian (turunan kedua).

$f_t^2(x_i)$: prediksi pohon pada iterasi ke-t.

$\Omega(f_t)$: Fungsi regularisasi pohon ke-t.

Pendekatan ini memungkinkan model untuk tidak hanya mengetahui arah penerunan error (gradien pertama), tetapi juga tingkat perubahan error (gradien kedua). Dengan demikian, proses optimasi menjadi lebih akurat dan efesien dibandingkan metode yang hanya mengunakan gradien pertama [52]. Secara praktis, hal ini membuat XGBoost lebih cepat dalam konvergensi dan mampu menemukan solusi optimal dengan baik

2.11.4 XGBoost untuk Klasifikasi Multi-Kelas

Dalam klasifikasi multi-kelas, XGBoost tidak memprediksi kelas secara langsung, melainkan melalui serangkaian perhitungan matematis yang terjadi pada setiap iterasi. Proses ini dimulai dari inisialisasi skor, konversi skor ke probabilitas menggunakan fungsi softmax, perhitungan *gradien* dan *hessian*, pemilihan split optimal, perhitungan bobot daun, hingga pembaruan skor model secara iteratif[50]. Berikut adalah uraian dari setiap tahapan tersebut.

1. Fungsi Softmax

   Dalam klasifikasi multi-kelas, XGBoost menghitung skor kumulatif $F^{(k)}(x_i)$ untuk setiap kelas k. Skor ini kemudian dikonversi menjadi probabilitas menggunakan fungsi softmax [54]:

   $$p_i^{(k)} = \frac{e^{F_t^{(k)}(x_i)}}{\sum_{j=0}^{C-1} e^{F_t^{(j)}(x_i)}} \quad (21)$$

   $Keterangan$:

   $p_i^{(k)}$: Probabilitas data ke- terhadap kelas k

   $F_t^{(k)}$: Skor kumulatif untuk kelas k pada iterasi ke-t

   C: Jumlah total kelas

   e: Bilangan Euler (≈ 2,71828)

2. Gradient (Turunan Pertama)

   Gradient mengukur arah kesalahan prediksi pada setiap iterasi. Untuk kasus klasifikasi multi-kelas, *gradient* dihitung berdasarkan turunan pertama dari fungsi *loss categorical cross-entropy* dengan *softmax*. Berdasarkan dokumentasi resmi XGBoost untuk pembuatan fungsi objektif multi-kelas.

   $$g_i^{(k)} = p_i^{(k)} - 1[y_i = k] \quad (22)$$

   Keterangan:

   $g_i^{(k)}$: Gradient untuk data ke-i pada kelas k

   $p_i^{(k)}$: Probabilitas prediksi untuk kelas k

   $1[y_i = k]$: Fungsi indikator

3. Hessian (Turunan Kedua)

   Hessian mengukur kecepatan perubahan *error* dan digunakan untuk mempercepat konvergensi. Keunggulan utama XGBoost dibandingkan metode *boosting* tradisional adalah penggunaan turunan kedua ini (Chen & Guestrin, 2016). Berdasarkan dokumentasi resmi XGBoost (XGBoost Contributors, 2024), hessian untuk klasifikasi multi-kelas dihitung sebagai:

   $$h_i^{(k)} = 2 \cdot p_i^{(k)} \cdot \left(1 - p_i^{(k)}\right) \quad (23)$$

Keterangan:

$h_i^{(k)}$: Hessian untuk data ke-i pada kelas k

$p_i^{(k)}$: Probabilitas prediksi untuk kelas k

4. Pemilihan Split (Gain)

Dalam membangun setiap pohon keputusan, XGBoost perlu menentukan percabangan (*split*) yang paling optimal. Pemilihan split didasarkan pada nilai gain, yaitu besarnya pengurangan *loss* yang diperoleh jika suatu *node* dipecah menjadi dua *node* anak [55]:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G_{all}^2}{H_{all}+\lambda}\right] - \gamma \quad (24)$$

Keterangan:

$Gain$: Pengurangan *loss* yang diperoleh dari suatu *split*

H_L, H_R: Total *hessian* pada *node* kiri dan *node* kanan

G_L, G_R: Total *gradien* pada *node* kiri dan *node* kanan

$H_{all}, G_{a_{ll}}$: Total *hessian* dan gradien sebelum split

λ: Parameter regularisasi L2 (default = 1)

γ: *Minimum loss reduction* atau biaya kompleksitas (default = 0)

5. Bobot Daun (*Leaf Weight*)

Setelah struktur pohon terbentuk dan data mencapai daun-daun tertentu, setiap daun akan diberikan bobot (*weight*) yang merepresentasikan kontribusi pohon tersebut terhadap prediksi akhir. Bobot daun dihitung berdasarkan total gradien dan hessian pada daun tersebut dengan mempertimbangkan parameter regularisasi [50].

$$w_j = -\frac{\sum_{i\in j} g_i^{(k)}}{\sum_{i\in j} h_i^{(k)}+\lambda} \quad (25)$$

Atau dapat ditulis dalam bentuk yang lebih ringkas:

$$w_j = -\frac{G_j}{H_j+\lambda} \quad (26)$$

Keterangan:

w_j: Bobot (output) dari daun ke-j

G_j: Total gradien dari semua data yang berada pada daun j

H_j: Total hessian dari semua data yang berada pada daun j

6. Pembaruan Skor Model

Tahapan terakhir dalam satu iterasi adalah memperbarui skor kumulatif model dengan menambahkan bobot daun yang telah dikalikan dengan *learning rate*. Proses ini merupakan inti dari mekanisme *boosting*, di mana setiap pohon baru secara perlahan memperbaiki kesalahan pohon sebelumnya [56].

$$F_1^{(k)}(x_i) = F_{t-1}^{(k)}(x_i) + \eta \cdot w^{(k)}(x_i) \quad (27)$$

Keterangan:

$F_1^{(k)}(x_i)$: Skor kumulatif untuk kelas k pada iterasi ke-t

$F_{t-1}^{(k)}(x_i)$: Skor kumulatif dari iterasi sebelumnya

η: *Learning rate* (laju pembelajaran, biasanya bernilai antara 0,1 hingga 0,3)

$w^{(k)}(x_i)$: Bobot daun tempat data x_i berada pada pohon kelas k.

2.11.5 Keunggulan XGBoost

XGBoost memiliki berbagai keunggulan yang menjadikannya salah satu algoritma paling populer dalam *machine learning*. Salah satu keunggulan utama adalah kemampuannya dalam menghasilkan akurasi yang tinggi dibandingkan algoritma lain seperti *Random Forest* dan *Gradient Boosting* tradisional [52]. hal ini disebabkan oleh penggunaan teknik optimasi yang lebih canggih serta kombinasi banyak model yang bekerja secara kolektif. Selain itu, XGBoost dilengkapi dengan mekanisme regularisasi yang efektif dalam mengurangi *overfitting*, sehingga model yang dihasilkan memiliki kemampuan generalisasi yang baik [50]. Dari sisi komputasi, XGBoost mendukung parallel *processing* yang memungkinkan proses pelatihan menjadi lebih cepat dan efisien, terutama pada dataset berukuran besar [50].

Keunggulan lainnya adalah kemampuannya dalam menangani *missing value* secara optimasi serta fleksibilitasnya dalam berbagai jenis permasalahan seperti klasifikasi, *regresi*, dan *ranking* [57]. Dengan berbagai keunggulan tersebut, XGBoost menjadi pilihan yang sangat tepat untuk digunakan dalam pengolahan data kompleks seperti fitur MFCC pada analisis musik tradisional.

2.12 Hyperparameter Optimization (HPO)

Dalam pengembangan model *machine learning* (ML), terdapat dua jenis parameter yang perlu dipahami secara fundamental, yaitu parameter model dan hyperparameter. Parameter model merupakan nilai yang dipelajari secara otomatis oleh algoritma selama proses pelatihan dari data, sedangkan hyperparameter adalah parameter yang ditentukan sebelum proses pelatihan dimulai dan mengatur perilaku serta efisiensi dari proses pelatihan itu sendiri [58]. *Hyperparameter Optimization* (HPO) didefenisikan sebagai proses pemilihan kumpulan hyperparameter yang paling efektif untuk

suatu algoritma pembelajaran [58]. Tujuan utama *Hyperparameter Optimization* adalah mengoptimalkan kinerja model dengan mencari konfigurasi hyperparameter yang menghasilkan performa prediksi terbaik pada data yang belum pernah dilihat sebelumnya. Proses optimasi hyperparameter sangat krusial karena pengaturan hyperparameter yang tepat dapat secara signifikan mempengaruhi kemampuan generalisasi model [59]. Begitu dengan sebaliknya pemilihan hyperparameter yang kurang tepat dapat menyebabkan model mengalami *under-fitting* atau *over-fitting*, yang berdampak pada buruknya performa model ketika dihadapkan pada data baru.

2.12.1 Bayesian Optimization

Bayesian Optimization Merupakan pendekatan optimasi yang di rancang untuk menemukan konfigurasi hyperparameter terbaik dari suatu fungsi objektif yang kompleks dan mahal secara komputasi [59]. *Bayesian Optimazition* bekerja secara iteratif dengan memanfaatkan informasi dari evaluasi hyperparameter sebelumnya untuk mengarahkan pencarian menuju area yang paling menjanjikan, sehingga mampu menemukan konfigurasi optimal dengan jumlah evaluasi yang relatif sedikit [59]. Algoritma *Bayesian Optimization* secara sistematis disajikan pada Gambar 2.3, yang diawali dengan pemilihan titik baru $Xn+1$ melalui optimasi *acquisition function* [$\alpha(X;Dn)$], kemudian mengevaluasi fungsi objektif untuk memperoleh nilai y_{n+1}, memperbarui data menjadi $Dn+1=\{Dn,(xn+1,yn+1)\}$, dan mengulangi proses ini secara iteratif hingga konvergensi [60].

Berikut adalah algoritma Bayesian Optimization [60].

1: **For** n = 1,2,...,**do**

2: Select new x_{n+1} by optimizing acquisition function α

$x_{n+1}=arg\ \text{x max}\ \alpha\ (X;Dn)$

3: Query objective function to obtain y_{n+1}

4: Augment data $Dn+1=\{Dn,(xn+1,yn+1)\}$

5: Update statistical model

6: **end for**

Dalam implementasinya, *Bayesian Optimization* dikembangkan ke dalam berbagai *framework*, salah satunya adalah Optuna.

2.12.2 Optuna

Optuna merupakan *framework* yang mengimplementasikan algoritma *Bayesian Optimization* dengan menggunakan *Tree-structured Parzen Estimator* (TPE) sebagai *surrogate model*-nya [58]. Berbeda dengan pendekatan *Bayesian Optimizatio* (BO) tradisional yang menggunakan *Gaussian Process*, TPE memodelkan distribusi nilai hiperparameter dengan membaginya menjadi dua kategori

berdasarkan ambang batas performa tertentu, yaitu distribusi untuk konfigurasi berperforma "baik" (l(p)) dan konfigurasi berperforma "buruk" (g(p)), sehingga proses pencarian menjadi lebih terfokus pada ruang parameter yang paling menjanjikan [59]. Pendekatan ini menawarkan fleksibilitas dan efisiensi yang lebih besar dibandingkan proses optimasi *Gaussian Bayesian* tradisional [61]. Dengan karakteristik tersebut, penerapan *Bayesian Optimization* melalui Optuna telah menjadi pendekatan standar dalam optimasi hiperparameter model *machine learning* untuk mencapai performa prediksi maksimal dengan beban komputasi yang minimal [61].

2.13 Confusion Matrix

Confusion matrix merupakan alat evaluasi performa yang digunakan untuk mengukur efektivitas suatu model klasifikasi dengan membandingkan nilai aktual (ground truth) terhadap nilai prediksi yang dihasilkan oleh model. Dalam permasalahan klasifikasi multikelas seperti penentuan asal daerah lagu, *confusion matrix* menyajikan ringkasan hasil prediksi dalam bentuk matriks $N \times N$, di mana N merepresentasikan jumlah kategori provinsi yang ada [62]. Bentuk umum confusion matrix multiclass dilihat pada Tabel 2.2.

Tabel 2. 2 Confusion Matrix Multiclass

Aktual/Prediksi	1	2	…	j
1	A_{11}	A_{12}	…	A_{1j}
2	A_{21}	A_{22}	…	A_{2j}
…	…	…	…	…
I	Ai_1	Ai_2	…	A_{ij}

Dalam konteks multikelas, kinerja model dievaluasi secara individual untuk setiap kelas i. Elemen-elemen dalam matriks diidentifikasi sebagai berikut :

1. True Positive (TP_i): Jumlah data yang secara aktual termasuk dalam kelas i dan berhasil diprediksi secara benar sebagai kelas i (terletak pada diagonal utama matriks).

2. False Positive (FP_i): Jumlah data dari kelas lain yang salah diprediksi oleh model sebagai kelas i. Nilai ini diperoleh dari jumlah seluruh elemen pada kolom i dikurangi nilai TP_i.

3. False Negative (FN_i): Jumlah data aktual kelas i yang salah diprediksi oleh model sebagai kelas lainnya. Nilai ini diperoleh dari jumlah seluruh elemen pada baris i dikurangi nilai TP_i.

Berdasarkan definisi tersebut, metrik evaluasi untuk performa sistem secara keseluruhan dapat dihitung menggunakan pendekatan *Macro-averaging*, yang memberikan bobot yang sama untuk setiap kelas tanpa memandang frekuensi kemunculannya di dataset :

1. Accuracy (Akurasi Global): Rasio total prediksi yang benar terhadap seluruh sampel data dalam dataset.

$$Accuracy = \frac{\sum_{i=1}^{n} TP_i}{Total\ Sampel} \quad (28)$$

2. Precision (Macro) : Rata-rata dari ketepatan prediksi di seluruh kelas. Menunjukkan seberapa akurat model saat memberikan prediksi.

$$Precision = \frac{1}{n}\sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i} \quad (29)$$

3. Recall (Macro): Rata-rata dari kemampuan model menemukan kembali label asli di seluruh kelas. Menunjukkan seberapa baik model dalam mengenali semua sampel lagu dari daerah tertentu.

$$Recall = \frac{1}{n}\sum_{i=1}^{n} \frac{TP_i}{TP_i + FN_i} \quad (30)$$

4. F1-Score (Macro): Rata-rata harmonis dari *precision* dan *recall* di seluruh kelas. Metrik ini menjadi indikator performa yang paling stabil untuk mengevaluasi model pada klasifikasi lagu daerah yang memiliki variasi karakteristik antar provinsi.

$$F1 - Score = 2 \times \frac{Precisin_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \quad (31)$$

1.14 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) adalah metode statistik untuk memvisualisasikan data berdimensi tinggi dengan menempatkan setiap titik data pada peta dua atau tiga dimensi [63]. t-SNE dikembangkan oleh van der Maaten dan Hinton pada tahun 2008 dan dipublikasikan dalam *Journal of Machine Learning Research* [63]. t-SNE merupakan algoritme iteratif yang memetakan titik-titik data ke ruang berdimensi rendah dengan mempertahankan struktur lokal data dari ruang dimensi tinggi asalnya. t-SNE merepresentasikan setiap objek sebagai sebuah titik pada *scatter plot* dua dimensi dan mengatur posisi titik-titik tersebut sedemikian rupa sehingga objek yang serupa dimodelkan oleh titik-titik yang berdekatan, sedangkan objek yang berbeda dimodelkan oleh titik-titik yang berjauhan [63].

Hasil visualisasi t-SNE dapat digunakan untuk menganalisis pola pengelompokan data apabila titik-titik dari kelas yang sama mengelompok dan terpisah dari kelas lain, hal tersebut mengindikasikan bahwa fitur yang digunakan memiliki kemampuan diskriminasi yang baik antar kelas. Sebaliknya, apabila titik-titik dari berbagai kelas bercampur, hal tersebut mengindikasikan adanya kemiripan representasi fitur antar kelas.[63].