Jurnal Elektronik Ilmu Komputer Udayana
Volume 14, No 2. November 2025

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Perbandingan Metode K-Nearest Neighbors Dan Support Vector Machine Pada Klasifikasi Lagu Daerah Dengan Penambahan Ekstraksi Fitur Principal Component Analysis

Roger Julian Sitorus[a1], I Gede Santi Astawa[a2] ,Luh Arida Ayu Rahning[a3], I Wayan Santiyasa[a4]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Badung, Bali, Indonesia
[1]roger.v.sitorus@gmail.com
[2]santi.astawa@unud.ac.id
[3]rahningputri@unud.ac.id
[4]santiyasa@unud.ac.id

Abstract

This study examines the application of the K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) algorithms in music data classification based on audio features on the IRSD (Indonesian Regional Songs Dataset) dataset. The main objective of this study is to evaluate the effect of using the Principal Component Analysis (PCA) dimension reduction technique on the performance of both algorithms. The experimental results show that the use of PCA with KNN improves classification accuracy, especially for the number of PCA components between 20 and 30 components. In the SVM model, the Radial Basis Function (RBF) kernel provides the highest accuracy, reaching 79%. Although PCA can reduce dimensions and speed up execution time, its effect on SVM is not significant. This study concludes that KNN with and without PCA can provide good classification results, while SVM with the RBF kernel is more stable and accurate. These findings provide insight for further research in the use of classification algorithms for more complex music audio data.

Keywords: *K-Nearest Neighbors, Support Vector Machine, Principal Component Analysis, Music Classification, Accuracy*

1. Pendahuluan

Indonesia merupakan sebuah negara dengan kekayaan budaya dan seni yang sangat berlimpah. Terdapat 38 provinsi yang tersebar dari Sabang sampai Merauke, masing-masing memiliki bentuk kesenian yang menjadi identitas khas daerahnya. Dalam berbagai medium kesenian, termasuk musik, setiap daerah di Indonesia menyimpan keunikan tersendiri yang membedakannya dari daerah lain. Lagu daerah merupakan salah satu bentuk ekspresi budaya yang tidak hanya mengandung nilai estetika, tetapi juga merepresentasikan nilai-nilai sosial, sejarah, serta bahasa lokal. Oleh karena itu, klasifikasi lagu daerah menjadi penting sebagai upaya pelestarian dan pengenalan budaya daerah secara lebih luas. Dengan melakukan klasifikasi, kita dapat membangun sistem yang mampu mengenali dan mengelompokkan lagu-lagu daerah berdasarkan ciri khasnya, sehingga dapat digunakan dalam pendidikan, pengarsipan digital, maupun sebagai bahan kajian dalam bidang etnomusikologi dan kecerdasan buatan.

KNN bekerja berdasarkan prinsip kedekatan antar data, yang membuatnya cocok digunakan pada dataset berukuran sedang seperti Indonesian Regional Song Dataset (IRSD). Dengan 68 fitur numerik yang merepresentasikan karakteristik audio, serta variasi genre berdasarkan 10 provinsi yang berbeda, KNN mampu menangkap pola-pola lokal antar lagu tanpa harus membangun model prediktif yang kompleks. Selain itu, dalam konteks data audio yang sering kali mengandung variasi alami maupun noise, KNN tetap dapat menghasilkan klasifikasi yang baik, terutama jika didukung dengan preprocessing seperti normalisasi data yang baik. Pada penelitian yang saya lakukan, dataset yang digunakan adalah musik dengan 10 kelas daerah provinsi, dimana variasi yang terdapat pada data tergolong cukup banyak, dihasilkan kesimpulan bahwa KNN dapat mengatasi masalah ini dengan baik tanpa komputasi yang rumit.

K-Nearest Neighbors (KNN) merupakan salah satu algoritma supervised learning dengan algoritma klasifikasi yang cukup sederhana dan tidak membutuhkan tuning hyperparameter maupun arsitektur yang kompleks. Dengan dataset yang tidak terlalu besar, akan lebih cocok untuk menggunakan KNN,

karena dengan arsitektur KNN yang sederhana memungkinkan pencegahan dari overfitting [1]. Terdapat penelitian yang telah dilakukan di tahun 2022, penelitian itu melakukan Klasifikasi Mood Musik dengan algoritma supervised learning KNN dan penggunaan ekstraksi fitur MFCC. Dataset yang digunakan memiliki 4 kelas mood, yaitu happy, sad, relax, dan angry. Dengan menggunakan 13 fitur dari hasil ekstraksi fitur MFCC, sehingga penelitian tersebut berhasil mendapatkan akurasi yang tergolong cukup memuaskan, yaitu 85,5% dengan algoritma KNN dan fitur ekstraksi MFCC serta metode Manhattan Distance [1].

Terdapat juga penelitian yang telah dilakukan pada 500 dataset lagu daerah dengan 10 kelas provinsi, dimana penulis tersebut menggunakan metode KNN dan SVM untuk melakukan komputasi pada klasifikasi lagu daerah dan proses waktu yang dibutuhkan untuk klasifikasinya tergolong cepat. Nilai K yang diambil oleh peneliti adalah 3 dan menggunakan 5-cross validation sebagai metode validasi. Hasil dari penelitian ini adalah akurasi yang didapatkan oleh algoritma KNN dan SVM ini tergolong kurang baik. Klasifikasi dengan KNN mendapatkan akurasi 69%, sedangkan SVM mendapatkan akurasi 73% [2]. Penelitian yang dilakukan oleh penulis menggunakan data sekunder, yaitu Indonesian Regional Song Dataset (IRSD) yang diperoleh dari situs Kaggle. Dataset IRSD ini merupakan hasil ekstraksi fitur pada Kumpulan lagu daerah berjumlah 500 lagu untuk 10 daerah provinsi, dimana fitur ekstraksi yang dihasilkan terdiri dari 68 fitur numerik dan terdapat juga 3 fitur kategorikal yang telah tersedia. Setiap daerah provinsi tersebut memiliki masing-masing 50 lagu yang akan digunakan pada pembangunan model klasifikasinya.

Berdasarkan rujukan dari beberapa penelitian diatas, penulis ingin melakukan implementasi pengklasifikasian lagu daerah berdasarkan kelas daerah provinsi dengan menggunakan algoritma KNN dan SVM pada data yang telah direduksi dengan Principal Component Analysis (PCA) untuk menghasilkan data dengan dimensi yang lebih sedikit. Penulis telah menentukan 10 kelas daerah yang digunakan pada penelitian yang terdiri dari Aceh, Jawa Barat, Jakarta, Jawa Tengah, Kalimantan Barat, Maluku, Papua, Riau, Sulawesi Utara, Sumatera Barat. Dengan pengklasifikasian ini, pengidentifikasian suatu lagu daerah diharapkan dapat menjadi lebih baik, sehingga lagu daerah yang berasal dari Indonesia dapat diketahui ciri khasnyya masing-masing. Penelitian ini akan menghasilkan suatu data uji yang didapatkan dari hasil pengujian pada model yang telah dibangun dengan metriks evaluasi yang digunkaan berupa akurasi, precision, recall dan F1-Score.

2. Metode Penelitian

Pada tahap awal, akan dilakukan pengumpulan data sekunder yang diperoleh dari website Kaggle. Dengan data yang telah diperoleh, akan dilakukan tahapan preprocessing dan juga tahapan PCA, yang merupakan salah satu metode tambahan untuk meningkatkan performa model KNN dan SVM. Setelah mendapatkan data hasil dari tahapan PCA, dilakukan penyimpanan terhadap data tersebut. Data yang telah berhasil disimpan akan diolah untuk digunakan dalam Pembangunan model klasifikasi menggunakan metode KNN dan SVM, yang bertujuan mencari titik terdekat sebuah track terhadap setiap class. Hasil dari model yang telah ditraining adalah komponen utama dari mesin klasifikasinya. Dengan model yang telah diperoleh tersebut, kita bisa melakukan testing. Evaluasi terhadap model juga akan dilakukan dengan mengambil nilai Precision, Recall, dan F1-Score. Hasil ini akan menjadi perbandingan performa dari KNN dan SVM dalam melakukan klasifikasi genre musik secara umum maupun terhadap lagu daerah.

2.1. Pengumpulan dan Load Data

Pada penelitian kali ini, penulis melakukan pengumpulan data sekunder. Data sekunder yang akan digunakan pada penelitian diperoleh dari website Kaggle. Nama dari dataset tersebut adalah IRSD: Indonesian Regional Song Dataset. IRSD adalah dataset musik berjumlah 500 lagu dengan 10 kelas daerah. Terdapat banyak sekali fitur-fitur yang tersedia dalam satu lagu daerah. Total jumlah fitur yang ada pada sebuah lagu adalah 67 fitur. Beberapa fitur yang ada pada dataset ini, yaitu Judul Lagu, Daerah, artist, Tempo, ZCR_mean, MFCC dan masih banyak lagi.

Data dimuat menggunakan pustaka pandas melalui fungsi pd.read_csv(). Dataset yang digunakan berformat CSV dan memuat sebanyak 500 entri yang berisi metadata lagu serta fitur-fitur numerik hasil ekstraksi sinyal audio seperti Zero Crossing Rate (ZCR_mean), Energy_mean, Spectral_Centroid_mean, dan lain-lain. Data ini juga mencakup label wilayah (region) sebagai target klasifikasi.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

2.2. Preprocessing Data

Tahap ini mencakup proses identifikasi dan penanganan data yang hilang (missing values), duplikat, atau nilai tidak valid. Data-data non-numerik yang tidak digunakan dalam klasifikasi seperti nama lagu dan nama artis dihapus. Proses ini bertujuan untuk meningkatkan kualitas data yang akan digunakan dalam pelatihan model.

Setelah data dibersihkan, langkah berikutnya adalah memisahkan fitur numerik dari data kategori lainnya. Fitur numerik ini yang akan digunakan sebagai input model. Sedangkan kolom target (region) yang menunjukkan wilayah asal lagu dipisahkan sebagai label yang akan diprediksi oleh model.

Fitur-fitur numerik selanjutnya dinormalisasi menggunakan StandardScaler dari Scikit-learn. Normalisasi ini sangat penting terutama karena model yang digunakan seperti SVM dan KNN sensitif terhadap skala fitur. StandardScaler bekerja dengan mengubah distribusi data agar memiliki rata-rata (mean) 0 dan standar deviasi 1, sehingga semua fitur berada dalam skala yang sebanding.

Setelah data ternormalisasi, dilakukan reduksi dimensi menggunakan metode Principal Component Analysis (PCA) dari Scikit-learn. PCA bertujuan untuk mereduksi kompleksitas data dengan mempertahankan variansi sebanyak mungkin dalam sejumlah kecil komponen. Seperti banyak metode multivariat, metode ini tidak digunakan secara luas hingga munculnya komputer elektronik, namun kini sudah tertanam dengan baik di hampir setiap paket komputer statistic [3]. Dalam penelitian ini, dipilih 10 komponen utama berdasarkan analisis explained variance ratio, yaitu komponen yang mampu merepresentasikan sebagian besar informasi dari data awal. Implementasi PCA membantu dalam mengurangi noise, mempercepat proses training, serta meningkatkan generalisasi model.

2.3. Splitting Training dan Testing Dataset

Setelah data siap, dataset dibagi menjadi dua bagian: data latih (training set) dan data uji (testing set) menggunakan train_test_split dari Scikit-learn. Pembagian data ini biasanya dilakukan dengan rasio 80:20, di mana 80% data digunakan untuk melatih model dan 20% sisanya digunakan untuk menguji performa model pada data yang belum pernah dilihat sebelumnya.

2.4. Pembuatan Model Klasifikasi KNN dan SVM

Pembangunan model KNN dimulai dengan menentukan parameter utama yaitu nilai K, yang merupakan jumlah tetangga terdekat yang akan digunakan untuk melakukan prediksi. Pada tahap training, model KNN tidak secara eksplisit membangun model dari data latih. Sebaliknya, data latih disimpan dan digunakan langsung saat proses klasifikasi. Saat model menerima data uji, ia menghitung jarak (misalnya, jarak Euclidean) antara data uji dan setiap titik data latih. Berdasarkan nilai K yang telah ditentukan, model akan memilih K tetangga terdekat dan menggunakan mayoritas label dari tetangga tersebut untuk menentukan genre dari lagu uji. Proses ini dilakukan berulang kali untuk setiap instance dalam data uji.

Pembangunan model SVM melibatkan pemilihan kernel yang tepat (linear, polynomial, RBF, dll.) untuk menangani data yang mungkin tidak terpisah secara linear dalam ruang fitur asli. Setelah fitur diekstraksi dan dataset dibagi, tahap training SVM dimulai dengan mencari hyperplane terbaik yang memaksimalkan margin antara kelas-kelas yang berbeda. SVM menggunakan teknik optimisasi untuk menemukan hyperplane ini. Jika data tidak dapat dipisahkan secara linear, kernel trick digunakan untuk memetakan data ke ruang berdimensi lebih tinggi di mana data menjadi lebih terpisah. Dalam proses ini, parameter regularisasi C juga disesuaikan untuk mengimbangi antara margin yang lebih lebar dan kesalahan klasifikasi yang lebih kecil.

2.5. Evaluasi Model

Setelah model dilatih, dilakukan evaluasi terhadap performa masing-masing model menggunakan metrik evaluasi seperti:

- Accuracy untuk melihat persentase prediksi yang benar secara keseluruhan,
- Confusion Matrix untuk memahami kesalahan klasifikasi pada tiap kelas wilayah,
- Precision, Recall, dan F1-Score untuk melihat keseimbangan performa model pada tiap kelas, khususnya bila data tidak seimbang.

Evaluasi ini bertujuan untuk memilih model terbaik yang dapat memprediksi asal wilayah lagu secara akurat berdasarkan fitur audio.

3. Hasil dan Pembahasan

3.1. KNN

Algoritma K-Nearest Neighbors adalah suatu metode algoritma klasifikasi yang bekerja berdasarkan tingkat kemiripan yang dihitung berdasarkan jarak (distance) terdekat dari data 46 pembelajarannya (data latih dan data uji) [4]. Metode yang paling banyak digunakan peneliti untuk menghitung jarak pada algoritma K-NN adalah metode Euclidean distance [5]. Implementasi dimulai dengan mendefinisikan fungsi perhitungan jarak Euclidean, yang merupakan metrik paling umum dalam algoritma KNN. Fungsi ini menghitung akar kuadrat dari jumlah kuadrat selisih antara setiap elemen fitur dua vektor, dan digunakan sebagai ukuran kedekatan antar titik dalam ruang fitur. Selanjutnya, sebuah kelas bernama KNN didefinisikan untuk merepresentasikan model, dengan parameter utama k, yaitu jumlah tetangga terdekat yang akan dipertimbangkan dalam proses klasifikasi.

Metode fit() dalam kelas ini bertugas menyimpan data latih X_train dan labelnya y_train. Data latih disimpan dalam bentuk DataFrame agar tetap kompatibel dengan struktur data pada proses selanjutnya, sedangkan label dikonversi menjadi array NumPy untuk efisiensi indexing. Seluruh proses prediksi dilakukan melalui metode predict(), yang menerima data uji dalam bentuk array berdimensi dua. Untuk setiap sampel uji, metode ini memanggil fungsi nearest_neighbors(), yang akan menghitung jarak Euclidean antara sampel uji dengan seluruh data latih, menyusun hasil berdasarkan jarak terdekat, dan memilih k label terdekat.

Proses prediksi kemudian dilakukan dengan menentukan label terbanyak (mayoritas) dari k tetangga tersebut. Hal ini sesuai dengan prinsip dasar KNN yang mengasumsikan bahwa data yang berada dalam jarak yang berdekatan cenderung memiliki kelas yang sama. Oleh karena itu, akurasi model KNN sangat bergantung pada kualitas preprocessing, termasuk normalisasi data agar setiap fitur memiliki skala yang seragam, serta penerapan dimensionality reduction seperti Principal Component Analysis (PCA) untuk mengurangi kompleksitas data dan mengatasi potensi overfitting akibat noise.

Dalam konteks penelitian ini, di mana fitur audio dari lagu-lagu daerah Indonesia memiliki dimensi yang tinggi dan keragaman yang besar, implementasi manual KNN memberikan keuntungan dalam hal transparansi proses, kemudahan modifikasi (misalnya untuk mengganti metrik jarak), serta kontrol penuh terhadap setiap tahapan klasifikasi. Kelebihan lain dari pendekatan ini adalah kemampuannya untuk digunakan sebagai dasar evaluasi komparatif terhadap algoritma klasifikasi lain yang lebih kompleks. Dengan demikian, algoritma KNN dapat menjadi pendekatan yang efektif dan efisien, khususnya pada data musik yang telah melalui tahapan preprocessing yang tepat.

3.1.1 KNN tanpa PCA

Nilai K	Precision	Recall	F1	Akurasi	Waktu Eksekusi
1	77%	78%	75%	79%	0.4s
3	76%	76%	73%	77%	0.4s
5	76%	75%	72%	75%	0.4s
7	74%	73%	68%	71%	0.5s
9	76%	73%	69%	73%	0.5s

Secara umum, terlihat bahwa nilai K yang lebih kecil (terutama K = 1) menghasilkan performa yang paling tinggi, dengan precision sebesar 77%, recall 78%, F1 score 75%, dan akurasi mencapai 79%, serta waktu eksekusi yang cepat (0,4 detik). Namun, model dengan K = 1 berpotensi mengalami

p-ISSN: 2301-5373
e-ISSN: 2654-5101

overfitting karena hanya mempertimbangkan satu tetangga, sehingga terlalu sensitif terhadap noise atau outlier pada data.

Ketika nilai K ditingkatkan ke 3 dan 5, performa model masih cukup baik dan relatif stabil. Metrik precision dan recall sedikit menurun, namun nilai F1 score dan akurasi tetap kompetitif, menunjukkan bahwa model mulai melakukan generalisasi yang lebih baik. Nilai K yang lebih besar (K = 7 dan 9) memperlihatkan penurunan performa secara bertahap. Misalnya, pada K = 7, F1 score turun menjadi 68% dan akurasi menjadi 71%, menunjukkan bahwa model mulai menyertakan tetangga yang mungkin kurang relevan dalam proses klasifikasi.

3.1.2 KNN dengan PCA

Nilai K	Jumlah Komponen PCA	Precision	Recall	F1	Akurasi	Waktu Eksekusi
1	10	60%	60%	58%	65%	0.5s
	20	71%	71%	69%	73%	0.5s
	30	72%	72%	70%	73%	0.6s
	34	72%	72%	70%	73%	0.5s
	38	72%	72%	70%	73%	0.5s
	40	72%	72%	70%	73%	0.5s
3	10	65%	64%	61%	67%	0.5s
	20	75%	75%	71%	74%	0.5s
	30	73%	72%	70%	74%	0.6s
	34	72%	70%	69%	74%	0.5s
	38	75%	76%	73%	77%	0.5s
	40	73%	73%	71%	76%	0.5s
5	10	64%	63%	60%	64%	0.5s
	20	73%	71%	69%	72%	0.6s
	30	74%	72%	70%	74%	0.5s
	34	78%	78%	74%	78%	0.5s
	38	78%	77%	73%	77%	0.5s
	40	78%	77%	73%	77%	0.5s
7	10	69%	68%	64%	68%	0.5s
	20	72%	70%	67%	70%	0.5s
	30	73%	72%	68%	72%	0.6s
	34	76%	74%	70%	74%	0.5s
	38	75%	74%	69%	73%	0.5s
	40	76%	74%	71%	74%	0.5s
9	10	70%	65%	61%	65%	0.5s
	20	75%	74%	69%	73%	0.5s
	30	77%	74%	70%	74%	0.5s
	34	77%	75%	71%	75%	0.5s
	38	77%	75%	71%	75%	0.5s
	40	76%	74%	70%	74%	0.5s

Secara umum, peningkatan jumlah komponen PCA dari 10 hingga sekitar 34 atau 40 menunjukkan perbaikan performa pada hampir semua nilai K. Misalnya, pada K = 1, precision meningkat dari 60% (dengan 10 komponen) menjadi 72% (dengan ≥30 komponen), dan akurasi dari 65% menjadi stabil di 73%. Hal ini menunjukkan bahwa PCA berhasil mereduksi dimensi sambil tetap mempertahankan informasi penting, yang meningkatkan kinerja klasifikasi.

Untuk K = 3, performa terbaik tercapai pada 38 komponen PCA dengan precision 75%, recall 76%, F1 score 73%, dan akurasi 77%. Tren serupa terlihat pada nilai K lainnya, di mana performa cenderung meningkat seiring bertambahnya jumlah komponen PCA, namun stabil setelah melewati sekitar 34 komponen. Ini menunjukkan bahwa menambah komponen lebih dari titik ini tidak memberikan peningkatan signifikan dan bahkan bisa menyebabkan overfitting.

Pada K = 5, kombinasi terbaik terjadi saat menggunakan 34 komponen PCA, menghasilkan precision dan recall sebesar 78% dan F1 score tertinggi yaitu 74%, dengan akurasi 78%. Nilai-nilai ini merupakan

yang tertinggi dalam keseluruhan tabel, menandakan bahwa K = 5 dan PCA 34 komponen adalah konfigurasi paling optimal untuk dataset ini.

Dari sisi waktu, seluruh eksperimen menunjukkan waktu eksekusi yang konsisten (0.5–0.6 detik), menunjukkan bahwa PCA tidak memberikan overhead signifikan dalam proses klasifikasi.

3.2. SVM

Implementasi model Support Vector Machine (SVM) dilakukan untuk skenario dengan dan tanpa PCA. Model menggunakan kernel Radial Basis Function (RBF), yang umum digunakan untuk masalah klasifikasi non-linear. Model dilatih dengan data latih dan diuji terhadap data uji menggunakan metode yang sama seperti KNN. Evaluasi performa dilakukan menggunakan metrik klasifikasi standar. Perbandingan hasil SVM dengan dan tanpa PCA memberikan pemahaman mendalam terkait dampak reduksi dimensi terhadap efektivitas klasifikasi wilayah berdasarkan fitur audio.

3.2.1 SVM tanpa PCA

Kernel	Precision	Recall	F1	Akurasi	Waktu Eksekusi
Linear	73%	73%	71%	75%	0.6s
Polynomial	67%	51%	51%	50%	0.6s
Sigmoid	67%	70%	67%	69%	0.5s
RBF	77%	77%	75%	79%	0.8s

Dari tabel diatas, dapat disimpulkan bahwa kernel RBF (Radial Basis Function) memberikan hasil terbaik secara keseluruhan, dengan precision dan recall masing-masing 77%, F1 score 75%, dan akurasi tertinggi sebesar 79%. Ini menunjukkan bahwa RBF sangat efektif untuk memetakan data non-linear ke dalam dimensi yang lebih tinggi sehingga memungkinkan pemisahan kelas yang lebih baik. Namun, kernel ini juga memerlukan waktu eksekusi paling lama (0.8 detik), yang merupakan kompromi yang wajar mengingat performanya.

Kernel linear memberikan hasil yang juga cukup baik, dengan precision dan recall 73%, F1 score 71%, dan akurasi 75%. Kernel ini cocok digunakan jika data relatif linear dan jumlah fitur besar, karena lebih sederhana dan lebih cepat daripada kernel non-linear.

Kernel sigmoid menghasilkan performa sedang, dengan precision 67%, recall 70%, F1 score 67%, dan akurasi 69%. Meski cukup seimbang, hasilnya masih di bawah linear dan RBF, menunjukkan bahwa sigmoid kurang optimal dalam memisahkan data pada kasus ini.

Sementara itu, kernel polynomial menunjukkan performa paling rendah, dengan precision dan recall masing-masing hanya 67% dan 51%, serta akurasi hanya 50%, yang mendekati hasil acak. Hal ini bisa disebabkan oleh overfitting atau ketidaksesuaian kompleksitas kernel polynomial terhadap distribusi data.

3.2.2 SVM dengan PCA

Kernel	Jumlah Komponen PCA	Precision	Recall	F1	Akurasi	Waktu Eksekusi
Linear	10	60%	60%	58%	65%	0.5s
	20	68%	68%	67%	71%	0.6s
	30	65%	65%	63%	68%	0.6s
	34	63%	63%	62%	69%	0.6s
	38	67%	68%	67%	72%	0.6s
	40	68%	68%	67%	72%	0.5s
Polynomial	10	65%	64%	61%	67%	0.5s
	20	73%	52%	52%	50%	0.5s

p-ISSN: 2301-5373
e-ISSN: 2654-5101

	30	73%	52%	53%	51%	0.6s
	34	72%	52%	53%	51%	0.6s
	38	72%	52%	53%	51%	0.6s
	40	72%	52%	53%	51%	0.6s
Sigmoid	10	64%	63%	60%	64%	0.5s
	20	63%	61%	59%	62%	0.6s
	30	62%	64%	61%	63%	0.6s
	34	60%	62%	60%	63%	0.5s
	38	63%	64%	62%	65%	0.6s
	40	63%	65%	63%	66%	0.6s
RBF	10	69%	68%	64%	68%	0.5s
	20	74%	74%	72%	77%	0.6s
	30	73%	73%	72%	77%	0.6s
	34	74%	74%	73%	78%	0.7s
	38	72%	73%	72%	77%	0.6s
	40	72%	73%	72%	77%	0.6s

Kernel RBF (Radial Basis Function) secara konsisten menghasilkan performa terbaik. Pada jumlah komponen PCA sebanyak 34, kernel ini mampu mencapai precision dan recall sebesar 74%, F1 score sebesar 73%, dan akurasi tertinggi yaitu 78%. Ini menunjukkan bahwa kernel RBF sangat efektif dalam menangani data yang kompleks dan non-linear, terutama setelah struktur data disederhanakan menggunakan PCA. Kernel linear menunjukkan peningkatan performa seiring bertambahnya jumlah komponen PCA. Dari precision awal 60% dengan 10 komponen, meningkat menjadi 68% pada 40 komponen, dengan akurasi tertinggi 72%. Ini menandakan bahwa PCA berhasil membantu kernel linear bekerja lebih baik dengan menyederhanakan distribusi fitur.

Sementara itu, kernel polynomial, meskipun sempat mencatat precision tinggi hingga 73%, justru gagal mempertahankan performa secara keseluruhan karena recall dan akurasinya rendah (sekitar 50–52%). Hal ini mengindikasikan terjadinya overfitting, di mana model hanya mengenali sebagian pola data namun gagal menggeneralisasi secara baik. Kernel sigmoid berada pada posisi menengah dengan performa yang stabil namun tidak menonjol; precision dan recall-nya berkisar antara 60–65%, dan akurasi tertinggi hanya 66%. Secara keseluruhan, dapat disimpulkan bahwa kombinasi kernel RBF dan jumlah komponen PCA antara 30 hingga 38 memberikan performa terbaik untuk klasifikasi, sementara kernel polynomial sebaiknya dihindari karena ketidakseimbangan kinerjanya. PCA terbukti efektif meningkatkan kinerja sebagian besar kernel, terutama yang berbasis non-linear.

4. Kesimpulan

Penelitian ini menunjukkan bahwa algoritma SVM dengan kernel RBF tanpa PCA memberikan performa terbaik secara keseluruhan, dengan akurasi tertinggi sebesar 79%, serta nilai precision, recall, dan F1-score masing-masing sebesar 77%, 77%, dan 75%. Ini menandakan bahwa model mampu mengklasifikasikan lagu daerah secara konsisten dan akurat ketika data tidak direduksi menggunakan PCA.

Namun, penerapan PCA tetap menunjukkan manfaat dalam beberapa konfigurasi, khususnya dalam KNN, di mana PCA dengan 34 komponen dan K=5 menghasilkan akurasi 78%, dengan precision dan recall sebesar 78%, serta F1-score sebesar 74%. Ini mengindikasikan bahwa reduksi dimensi dengan PCA dapat meningkatkan performa KNN, terutama dalam hal kecepatan eksekusi dan generalisasi model, tanpa mengorbankan akurasi secara signifikan.

Sementara itu, SVM dengan kernel polynomial dan sigmoid tidak memberikan hasil yang memuaskan, bahkan setelah penerapan PCA, dengan akurasi berkisar di antara 50–69%, serta nilai F1-score yang relatif rendah. Hal ini menunjukkan bahwa jenis kernel tersebut kurang cocok untuk pola data lagu daerah yang digunakan dalam penelitian ini.

Penelitian selanjutnya disarankan untuk menambah jumlah dan keberagaman data lagu daerah guna meningkatkan representasi budaya dan mengurangi bias model. Selain itu, tahapan pembersihan data dan pengurangan noise perlu diperhatikan agar hasil klasifikasi lebih akurat. Penggunaan metode lanjutan seperti ensemble learning (misalnya Random Forest atau Gradient Boosting) serta eksplorasi model berbasis deep learning juga dapat dipertimbangkan untuk meningkatkan performa klasifikasi.

Dengan langkah-langkah tersebut, diharapkan hasil klasifikasi dapat menjadi lebih andal dan aplikatif dalam pelestarian musik tradisional Indonesia.

Referensi

[1] F. F. Surenggana., A. Aranta., F. Bimantoro. 2022. “Klasifikasi Mood Musik Menggunakan K-Nearest Neighbor Dengan Mel Frequency Cepstral Coefficients”. Jurnal Teknologi Informasi, Komputer, dan Aplikasinya Universitas Mataram, vol. 4, no. 2, pp.

[2] Gst. A.V.M.Giri., M.L. Radhitya. 2023. “Klasifikasi Lagu Daerah di Indonesia dengan Metode Machine Learning”. Jurnal Nasional Teknologi Informasi dan Aplikasinya, vol. 1, no. 3, pp,

[3] Jolliffe, I.T. (2022). Principal Component Analysis: Second Edition. New York: Springer.

[4] Boiculese, V.L., Dimitriu, G. and Moscalu, M. 2013. “Improving Recall Of K-Nearest Neighbor Algorithm For Classes Of Uneven Size”, E-Health and Bioengineering Conference 2013, (1), pp. 23–26. doi:10.1109/EHB.2013.6707403.

[5] Bing, L. 2012. “A SURVEY OF OPINION MINING AND SENTIMENT ANALYSIS”. doi:10.1007/978-1-4614-3223-4.