**Klasifikasi Musik Berdasarkan *Genre* pada Layanan Musik *Streaming* Musik Spotify menggunakan Algoritma *K–Nearest Neighbor* dan *Modified K–Nearest Neighbor***

**I Made Tangkas Wahyu Kencana Yuda1) I Wayan Supriana2) I Gede Santi Astawa3)**

Program Studi Teknik Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam1) 2) 3)

Universitas Udayana, Badung, Bali1) 2) 3)

[tangkaswahyu98@gmail.com](mailto:tangkaswahyu98@gmail.com) 1) wayan.supriana@unud.ac.id2) santi.astawa@unud.ac.id3)

# ***Abstract***

*Music is the science and art of arranging tone or sounds in sequence to produce sound compositions that have unity & balance. Music is divided into several different genres. Several studies have been carried out, there are various methods that are often used to classify music genres, one of which is the KNN. This study uses 12 audio features with 14 genres on Spotify to determine the effect of song attributes used on performance to classify songs and the process of comparing two methods, namely K-Nearest Neighbor (KNN) & Modified K-Nearest Neighbor (MKNN) using k-Fold cross-validation & confusion matrix. In addition, the PCA method will also be used to select features. From the research conducted, it was found that the MKNN was less effective than the KNN. By an accuracy difference of 1.1% at the optimal k ratio and optimal PCA results in an accuracy of 1.0%. Whereas the KNN method without PCA at the optimal k ratio with the KNN method using PCA at the optimal k ratio and the optimal PCA having an accuracy difference of 0.4%. Therefore, the KNN method without PCA based on its level of accuracy is the best method in this study.*

***Keyword*** *:* *K–Nearest Neighbor, Modified K–Nearest Neighbor, Classify Music*

# **PENDAHULUAN**

Menurut KBBI tahun 2002, musik mempunyai arti bahwa ilmu dan seni menyusun nada/suara dalam urutan, kombinasi, dan hubungan temporal untuk memperoleh suara yang memiliki kesatuan dan kesinambungan. Selanjutnya musik memiliki arti, bahwa nada/suara yang disusun sedemikian rupa sehingga mengandung irama, lagu, dan keharmonisan (Setiawan, 2020). Perkembangan teknologi dalam era modern ini, musik dapat didengarkan melalui berbagai macam media, baik melalui media transmisi radio hingga media *streaming*. Adapun jasa layanan *streaming* *music*, diantaranya Spotify, Joox, Apple Music, Deezer dan lain sebagainya. Sejak diluncurkan pada 2008 lalu, Spotify merupakan salah satu jasa layanan *streaming music* digital yang paling banyak digunakan oleh pendengar musik (Iriansyah, 2018). Pengguna Spotify dapat mengakses musik berdasarkan *artist*, album, *genre*, *playlist*, *podcasts*, atau label rekaman (Spotify, 2020).

Menurut Giri (2018), *genre* musik adalah salah satu cara pengelompokan musik sesuai dengan kemiripannya satu sama lain yang sangat umum digunakan untuk mengatur *database* musik digital. *Database* atau katalog musik yang diorganisasi berdasarkan *genre* musik memberikan kemudahan pada pendengar musik untuk mencari musik–musik sejenis yang sesuai dengan referensinya. Bertambahnya jumlah musik dalam bentuk digital secara pesat menyebabkan pemberian label *genre* secara manual menjadi tidak efektif. Pemberian label *genre* secara otomatis dapat dilakukan dengan menerapkan algoritma kecerdasan buatan yang dapat mengklasifikasi musik berdasarkan *genre* dengan menggunakan fitur–fitur pada atribut lagu. Terdapat banyak *genre* musik seperti *Jazz, Gospel, Blues, Funk, Rock, EDM, Reggae, Hip–Hop, Pop, Pop Punk, Pop Rock, Slow Pop*, dan lainnya. Dari berbagai *genre* musik tersebut kemudian terbagi lagi menjadi beberapa *playlist*. Pembagian *genre* bertujuan untuk pengguna memilih lagu kesukaannya sesuai *genre*.

Data musik terdiri dari berbagai fitur. Data musik yang berdimensi tinggi merupakan salah satu kendala dalam penerapan teknik machine learning karena akan memberikan efek negatif terhadap proses analisis. Untuk menangani data musik berdimensi tinggi tersebut, mereduksi fitur menjadi hal yang sangat penting. Dengan pengurangan fitur sangat penting untuk mengidentifikasi faktor risiko paling signifikan. Maka diperlukan sebuah tahapan seleksi fitur untuk memilih fitur-fitur mana saja yang digunakan dalam penelitian, agar dapat mencapai nilai akurasi klasifikasi berdasarkan genre yang lebih tinggi. Pada penelitian ini akan menggunakan seleksi fitur *Principal Component Analysis* (PCA). PCA merupakan salah satu teknik statistik terkenal yang bertujuan untuk mengurangi dimensi data tanpa kehilangan informasi penting dalam data (Dananjaya, 2019).

Terdapat beberapa penelitian sebelumnya mengenai pengklasifikasian pada *genre* dan fitur musik. Antara lain, pada penelitian tugas akhir oleh Supriyadi (2018) menggunakan lagu dengan 2 *genre* musik yaitu *pop* dan klasik dengan metode *Backpropagation* dan menggunakan 7 fitur audio yaitu *accousticness*, *danceability*, *energy*, *loudness*, *speechiness*, *tempo* dan *valence.* Pada layanan *streaming music* Spotify dengan menerapkan *Artificial Neural Network* (ANN). Hasil akurasi dari pengujian diperoleh sebesar 99,5%.

Pada penelitian sebelumnya, Giri (2018), melakukan penelitian klasifikasi musik berdasarkan 10 *genre* yaitu *classical, EDM, hip–hop, metal, pop, punk, R&B, rap, soul* dan *rock* dengan metode *K–Nearest Neighbor (KNN)* dan menggunakan 11 fitur audio pada atribut lagu (*speechiness, energy, danceability, loudness, tempo, mode, valence, instrumentalness, accousticness, key,* dan *liveness*). Penelitian ini mendapatkan nilai akurasi klasifikasi KNN sebesar 44,8%. Nilai akurasi *genre* yang tertinggi ada pada *genre Classical* yaitu 100%, sedangkan nilai akurasi *genre* yang terendah ada pada *genre* *Pop* yaitu 25%.

KNN terdapat permasalahan outlier untuk perhitungan bobot. Sehingga untuk mengatasi hal tersebut Parvin, dkk menggunakan *Modified K–Nearest Neighbor* (MKNN) sebagai pembanding dan didapatkan hasil yang menggunakan metode MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan KNN. Akurasi terendah dan tertinggi dari MKNN secara berurutan yaitu 62.52% dan 95.90%. Nilai akurasi terendah dan tertinggi keduanya berada pada nilai *k* = 5. Sehingga diperlukan nilai *k* yang optimum, maka didapatkan akurasi yang tinggi dengan rentang nilai yang masih berdekatan. MKNN mampu menambah nilai akurasi dari metode klasifikasi KNN (Parvin, dkk, 2010).

Perbandingan metode KNN dan MKNN tidak hanya dilakukan oleh Parvin tetapi juga oleh Okfalisa dengan hasil sebelum klasifikasi, *k–fold cross validation* dilakukan untuk mencari pemodelan data yang optimal menghasilkan pemodelan data pada *cross* 2 dengan akurasi 93,945%. Hasil pemodelan *k–fold cross–validation* akan menjadi model untuk sampel data pelatihan dan pengujian data untuk menguji KNN dan MKNN untuk klasifikasi. Hasil klasifikasi menghasilkan akurasi berdasarkan aturan *confusion matrix*. Tes menghasilkan akurasi tertinggi KKN sebesar 94,95% dengan akurasi rata–rata selama tes adalah 93,94% dan akurasi tertinggi MKNN adalah 99,51% dengan akurasi rata–rata selama tes adalah 99,20% (Okfalisa, dkk, 2017). Selain Okfalisa, dkk, perbandingan KNN dan MKNN juga dilakukan oleh Ravi dengan hasil membuktikan bahwa metode MKNN cenderung lebih tinggi akurasinya dibandingkan dengan metode KNN dimana metode MKNN memiliki tingkat akurasi 76,66% sedangkan KNN 43,33% (Ravi, dkk, 2019).

Dari penelitian–penelitian yang sudah ada tersebut. Penelitian ini menggunakan 12 fitur audio dengan 14 *genre* pada Spotify dengan membandingkan kedua metode algoritma klasifikasi *K–Nearest Neighbor* (KNN) dengan metode algoritma klasifikasi *Modified K–Nearest Neighbor* (MKNN). Kemudian pada penelitian ini juga akan menggunakan seleksi fitur menggunakan *Principal Component Analysis* (PCA) dengan membandingkan kedua metode algoritma klasifikasi *K–Nearest Neighbor* (KNN) dengan metode algoritma klasifikassi *Modified K–Nearest Neighbor* (MKNN).



# **TINJAUAN PUSTAKA**

## **Identifikasi Fitur Audio Spotify**

Untuk setiap lagu di platform mereka, Spotify menyediakan data untuk 13 fitur audio pada atribut lagu antara lain, *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode, duration* dan *time\_signature* (Ashrith, 2018).

Pada penelitian ini akan menggunakan variabel fitur audio pada atribut lagu Spotify yang digunakan adalah 12 fitur antara lain *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode* dan *duration* yang diambil dengan menggunakan Spotify API.

## **Transformasi Data**

Menurut Supriyadi (2018), proses transformasi dilakukan untuk menskalakan atribut numerik dalam *range* yang lebih kecil, seperti –1.0 sampai 1.0 atau 0.0 sampai 1.0. Ada banyak metode transformasi, salah satunya *min–max normalization*. *Min–max normalization* merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli, yang dituliskan dalam pada Persamaan (2.1).

(2.1)

Keterangan:

*V’* = Nilai yang dicari.

*V* = Nilai pada variabel A.

𝑀𝑖𝑛𝐴 = Nilai minimal variabel A.

𝑀𝑎𝑥𝐴 = Nilai maksimal variabel A.

𝑁𝑒𝑤\_𝑚𝑖𝑛𝐴 = Rentang nilai minimal pada variabel A.

𝑁𝑒𝑤\_𝑚𝑎𝑥𝐴 = Rentang nilai maksimal pada variabel A.

## **Principal Component Analysis**

Menurut Jolliffe (2002), *Principal Component Analysis* (PCA) merupakan metode untuk mengurangi dimensi kumpulan data yang terdiri dari sejumlah besar variabel yang saling terkait, sambil mempertahankan sebanyak mungkin variasi yang ada dalam kumpulan data. Tujuan dari pca adalah mentransformasikan ke satu set variabel baru, mengecek korelasi antar variabel data dan mempertahankan sebagian besar variasi yang ada di semua variable asli. Dengan melihat korelasi antar variabel tersebut akan diperoleh faktor*–*faktor yang mempengaruhi pada penelitian ini. Langkah*–*langkah PCA digunakan sebagai berikut (Dananjaya, 2019).

1. Menghitung *variance*

Menghitung *variance* menggunakan rumus pada Persamaan (2.2).

(2.2)

1. Menghitung *covariance*

Menghitung *covariance* menggunakan rumus pada Persamaan (2.3). Setelah itu dihasilkan *covariance matrix*.

(2.3)

1. Menghitung *eigenvalue* dan *eigenvectors*

Menghitung *eigenvalue* dan *eigenvector* pada Persamaan (2.4). Sebelum menghitung *eigenvalue* dan *eigenvector*, nilai lamda harus dihitung terlebih dahulu pada Persamaan (2.5).

(2.4)

(2.5)

1. Mengurutkan *eigenvalue* dan *eigenvectors*, setelah itu tentukan *principical component* (komponen utama).
2. Menghitung korelasi antara variabel utama dengan *principal component* (komponen utama)

Setelah menentukan komponen utama, selanjutnya mengkorelasikan komponen utama dengan variabel utama akan dihitung pada Persamaan (2.6). Mengurangi variabel dengan menghilangkan komponen rendah.

(2.6)

Keterangan:

: *Matrix* n x n.

: *Eigenvalue*.

: *Identity Matrix*.

: Korelasi antara variabel utama dengan komponen utama.

: *eigenvectors*.

: *Covariance Matrix*.

## **K–Fold Cross Validation**

*Cross–validation* (CV) adalah metode statistik yang dapat digunakan untuk mengevaluasi kinerja model atau algoritma dengan membagi sampel data secara acak dan mengelompokkan data menjadi dua *subset* yaitu data proses pembelajaran dan data validasiatauevaluasi. Model atau algoritma dilatih oleh *subset* pembelajaran dan divalidasi oleh *subset* validasi. *K–Fold Cross–Validation* adalah kasus khusus dari *cross–validation* di mana mengulangi set *dataset* *k* kali. Dalam setiap putaran, *dataset* akan dibagi menjadi bagian *k* satu bagian digunakan untuk validasi, dan bagian yang tersisa digabung ke dalam *subset* *training* (data latih) untuk evaluasi. *Ten–fold cross****–****validation* adalah salah satu *k–fold cross–validation* yang direkomendasikan untuk pemilihan model terbaik (Wibowo, 2017).

## **Confusion Matrix**

*Confusion Matrix* adalah tabel yang sering digunakan untuk menggambarkan kinerja model pada klasifikasi (atau "*classifier*") pada serangkaian data *testing* (data uji) yang nilai sebenarnya diketahui (Nomleni, 2015). Tabel 1 menunjukkan *confusion matrix* untuk *classifier*.

**Tabel 1.** *Confusion Matrix*

|  |  |  |
| --- | --- | --- |
| Data Prediksi | Data Aktual | |
| Ya | Tidak |
| Ya | TP | FN |
| Tidak | FP | TN |
| Total | P | N |

Pada penelitian ini, entri dalam *confusion matrix* memiliki arti seperti berikut ini:

* + TP adalah *true positive*, merupakan jumlah data prediksi yang benar diklasifikasikan *positif*.
  + TN adalah *true negatif*, merupakan jumlah data prediksi yang benar diklasifikasikan *negatif*.
  + FN adalah *false negatif*, merupakan jumlah data prediksi yang salah diklasifikasikan *negatif*.
  + FP adalah *false positif*, merupakan jumlah data prediksi yang salah diklasifikasikan *positif*.

Pada penelitian ini ukuran kinerja dari sistem klasifikasi yaitu akurasi pada Persamaan (2.7) untuk mengukur seberapa tepat klasifikasi terhadap kelas. Akurasi adalah rasio untuk mengukur kedetakan antara prediksi benar (*positif*) dengan aktual (sebenarnya) dari keseluruhan data.

(2.7)

## **K–Nearest Neighbor**

Metode Algoritma *K–Nearest Neighbor* (KNN) yaitu algoritma yang bertujuan untuk klasifikasi data. Secara umum, metode ini digunakan untuk menentukan kelas data dari suatu data dengan mengukur jarak antar tetangganya (Parsian, 2015). Jarak yang dipilih adalah jarak terpendek dari data *testing* (data uji). Perbandingan jarak antara data *testing* (data uji) dan data *training* (data latih) dilakukan sebanyak *k* buah. Rumus perhitungan jarak tersebut dapat dilakukan dengan *euclidean distance* dituliskan pada Persamaan (2.8).

(2.8)

Keterangan:

*Euclidean Distance()* : Jarak antara data latih dengan data uji.

: Atribut fitur data latih.

: Atribut fitur data uji.

*i* : Mempresentasikan index nilai atribut fitur.

*n*  : Jumlah panjang dimensi atribut fitur.

Dalam menghitung *euclidean distance*, terdapat beberapa Langkah yang dikerjakan sebelum dan setelah pengukuran jarak antar data latih dengan data uji antara lain:

* Langkah pertama, yang harus dilakukan adalah menentukan nilai dari *k*, nilai ini tergantung pada data dan persyaratan dari masalah yang diselesaikan.
* Langkah kedua, dilakukan perhitungan jarak antara data latih dengan data uji.
* Langkah ketiga, mengurutkan nilai dari jarak–jarak yang didapatkan dari nilai terendah hingga tertinggi, kemudian dipilih sebanyak *k* buah nilai.
* Langkah keempat, nilai kelas data dari *k* buah tersebut dikumpulkan.
* Langkah kelima, yaitu penentuan yang dilakukan dengan *vote* jumlah kelas data terbanyak sebagai kelas data dari data uji.

Keunggulan dari algoritma *K–Nearest Neighbor* ini adalah tidak memerlukan waktu yang lama untuk proses data data *training* (data latih), jika dibandingkan dengan algoritma *Backpropagation* (Redjeki, 2013).

## **Modified K–Nearest Neighbor**

Metode Algoritma *Modified K–Nearest Neighbor* (MKNN) merupakan algoritma pengembangan dari kelemahan algoritma *K–Nearest Neighbor* (KNN). Kelemahan *K–Nearest Neighbor* (KNN) terdapat pada data *training* (data latih) dan data *testing* (data uji) yang hanya berbasis jarak untuk menentukan ketetanggan pada setiap data *training* (data latih) dan data *testing* (data uji). Secara umum gagasan metode algoritma *Modified K–Nearest Neighbor* (MKNN) yaitu dengan menetapkan sebuah label kelas data dari sejumlah *k* data dari data *training* (data latih) dengan memperhitungkan nilai validitas dan *weight voting* (bobot) (Parvin dkk, 2010). Adapun beberapa tahapan dari metode ini yaitu:

* Tahap pertama, menentukan jarak antara data latihdengan data ujipada Persamaan (2.8). Kemudian diurutkan dan dipilih sebanyak *k* yang terpendek.
* Tahap kedua, dari setiap data latihterpendek ditentukan nilai validitasnya. Nilai validitas ditentukan dari data latihlainnya yang terdekat. Jumlah seluruh validitas dibagi dengan *k*. Rumus nilai validitas pada Persamaan (2.4).

(2.9)

Keterangan:

: Nilai validitas.

: Besaran atau jumlah titik terdekat dari individu.

: Kelas data latih.

: Kelas data latih lainnya ke–i yang terdekat dengan .

Fungsi *S* merupakan indeks kesamaan dari 2 buah data latih. Menentukan nilai *S* yang dituliskan pada Persamaan (2.5). Jika antar 2 data latih memiliki kelas data yang sama dengan data uji, maka nilai *S* yaitu 1. Sedangkan jika antar 2 data latih memiliki kelas data yang berbeda dengan data uji, maka nilai *S* yaitu 0.

(2.10)

Keterangan:

*a* : Kelas *a* pada data latih.

*b* : Kelas lain selain *a* pada data latih.

* Dari sejumlah nilai *k* yang terpendek didapat. Lalu dilakukan proses mencari *weight* (bobot) pada Persamaan (2.6). Nilai distance ditambah dengan 0,5 untuk menghindari penyebut yang bernilai 0. Jika penyebut bernilai 0, *weight* (bobot) akan menghasilkan nilai tidak terdefinisi (Parvin dkk, 2010).

(2.11)

Keterangan:

: Bobot data latih ke–i.

: Validitas data latihke–i.

: Jarakantara data latihke–i dengan data uji.

: *Smooting* (pemulusan) bernilai 0,5.

* Dilakukan pemilihan (*voting*) kelas data yang akan menjadi kelas data dari data uji. Pemilihan tersebut dengan menjumlahkan seluruh *weight* (bobot) pada masing–masing kelas data, sesuai dengan jumlah nilai *k* yang digunakan. Sehingga nilai *voting* dari kelas data terbesar, merupakan hasil kelas data dari data uji.

(2.12)

Keterangan:

: *Voting* dari kelas data latih x.

: Jumlah data latihdengan kelas data x.

: Nilai indeks.

: Bobot data latihke–i.

# **Metode Penelitian**

## **Dataset**

Secara keseluruhan, terdapat 18 variabel atribut lagu dalam API Spotify. Namun hanya 13 variabel yang digunakan pada penelitian ini, yaitu 1 fitur pengguna (*playlist*) dan 12 fitur audio (*danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode,* dan *duration*).

Data musik yang dipergunakan sebagai dataset dalam penelitian ini berasal dari layanan *streaming music* Spotify, berjumlah 50 musik untuk setiap genre. Genre yang digunakan berjumlah 14 *playlist*, yaitu *Blues, Classical, Country, Gospel, Hip Hop, Jazz, Metal, Pop, Reggae, Rock, EDM, Funk, R&B,* dan *Soul*. Jadi, total musik yang digunakan dalam penelitian ini adalah 700 musik. Masing*–*masing 14 *playlist* yang telah dibuat, akan di beri label pada *playlist* tersebut, agar bisa membedakan antar *genre*. Untuk *genre* *Blues* akan diberi label 0, *genre* *Classical* diberi label 1, *genre* *Country* diberi label 2, *genre* *EDM* diberi label 3, *genre* *Funk* diberi label 4, *genre* *Gospel* diberi label 5, *genre* *Hip Hop* diberi label 6, *genre* *Jazz* diberi label 7, *genre* *Metal* diberi label 8, *genre* *Pop* diberi label 9, *genre* *R&B* diberi label 10, *genre* *Reggae* diberi label 11, *genre* *Rock* diberi label 12, dan *genre* *Soul* diberi label 13.

## **Pengujian Pertama**

Pada pengujian pertama ini akan membandingkan akurasi dari tiap perubahan nilai *k* pada *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan metode *k–fold cross–validation* dengan nilai *k* adalah 10. Data yang digunakan sebanyak 700 data lagu, dengan pembagian jumlah data yaitu 90% data *training* (data latih) sebanyak 630 data lagu dan 10% data *testing* (data uji) sebanyak 70 data lagu. Untuk 70 lagu data *testing* (data uji) akan diambil secara merata yaitu 5 data lagu pada tiap–tiap *genre* atau tiap kelas data. Sehingga, data dipartisi menjadi 10 bagian dengan masing–masing partisi berisi 70 data. Sebelum melakukan klasifikasi pada 700 data lagu akan di acak terlebih dahulu, agar tidak sesuai urutan tiap *genre* nya.

## **Pengujian Kedua**

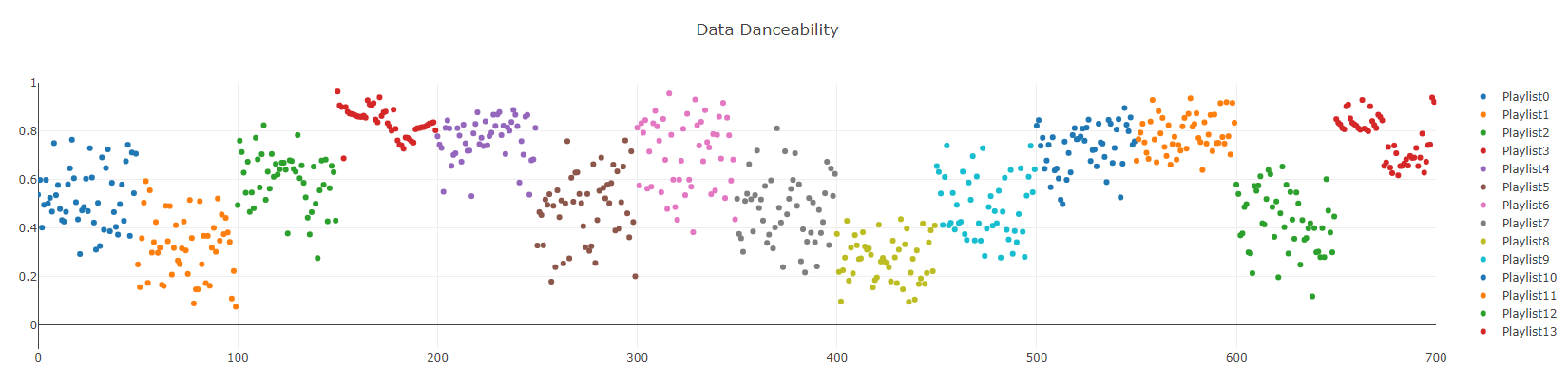
Pada pengujian kedua ini, akan dilakukan sama seperti pengujian pertama, tetapi data *genre* pada pengujian pertama akan direduksi menggunakan metode seleksi fitur *Principal Component Analysis* (PCA) menjadi variabel fitur baru dengan cara menghitung nilai *variance* pada Persamaan (2.2), menghitung nilai *covariance* pada Persamaan (2.3), setelah itu menghitung *eigenvalue* dan *eigenvector* pada Persamaan (2.4). Sebelum menghitung *eigenvalue* dan *eigenvector*, nilai dihitung terlebih dahulu pada Persamaan (2.5), selanjutnya mengurutkan *eigenvalue* dan *eigenvector* serta menentukan komponen utama. Kemudian melakukan korelasi antara variabel fitur utama dengan komponen utama pada Persamaan (2.6).

Jumlah variabel fitur baru yang akan diuji yaitu 1 sampai dengan 11 fitur, selanjutnya fitur baru tersebut akan diklasifikasi dengan membandingkan akurasi tertinggi dari tiap perubahan nilai *k* pada tiap*–*tiap variabel fitur baru dengan *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan metode *k–fold cross–validation* dengan nilai *k* adalah 10.

# **Hasil Penelitian**

## **Pengaruh Atribut Lagu yang digunakan terhadap Kinerja untuk mengklasifikasi Lagu menggunakan KNN dan MKNN**

Dataset yang telah terkumpul terdiri dari 700 musik dari 14 genre berbeda. Tiap data musik terdiri dari 12 fitur audio. Gambar 1 menunjukkan plot sebaran data dilihat dari *Danceability*, fitur-fitur audio memiliki sebaran data yang berbeda tiap kelas data nya. Kelas data *genre EDM,* *Reggae*, *Soul* dan *Funk* berdasarkan variabelyang digunakantrek nya lebih cocok menari (*Danceability)*, musik *Funk* memiliki emosi yang bervariasi sementara musik *Blues* cenderung ke emosi negatif, musik rock dan metal lebih energik, musik metal memiliki tempo cepat, musik *Classical* dan *Jazz* lebih lembut dan menenangkan, musik *Hip Hop* trek musik dan ucapan vokal nya seperti musik lagu rap, musik *Classical* dan *Jazz* treknya tidak ada ucapan vokal, musik *Gospel* trek nya lebih live, musik *Classical*, *Jazz* dan *Pop* karena treknya lebih akustik, musik *Classical* dan *Funk* memiliki durasi lagu terlama, dan semua *genre* yang digunakan pada penelitian ini menggunakan trek kunci tangga nada seperti minor dan mayor.



**Gambar 1.** Plot sebaran *Danceability*

## **Hasil Pengujian Pertama dan Kedua**

Dari keseluruhan nilai *k* pada data uji klasifikasi *genre*. Pada pengujian pertama, KNN lebih efektif dibandingkan dengan MKNN pada klasifikasi musik berdasarkan *genre* menggunakan 10*–fold cross–validation*. Hal ini terlihat dari keseluruhan nilai *k* pada data uji klasifikasi *genre*, metode MKNN memiliki akurasi yang lebih rendah dibandingkan dengan metode KNN. Pada nilai *k* = 9 dan 11 dengan nilai akurasi 60.7% merupakan akurasi tertinggi pada metode KNN, sedangkan pada metode MKNN memiliki nilai akurasi tertinggi sebesar 59.6%, pada nilai *k* adalah 9.

Sedangkan pada pengujian kedua menggunakan seleksi fitur *Principal Component Analysis* (PCA), PCA-KNN lebih efektif dibandingkan dengan PCA-MKNN pada klasifikasi musik berdasarkan *genre* menggunakan 10*–fold cross–validation*. Hal ini terlihat perbandingan *k* yang paling optimal dan PCA yang paling optimal dari akurasi PCA yang tertinggi pada PCA 11 fitur akurasi tertinggi dari PCA-MKNN yaitu 59.3% dengan nilai *k* = 9, sedangkan akurasi dari PCA-KNN saat nilai *k* = 9 yaitu 60.3%.

**Gambar 2.** Perbandingan Akurasi MKNN dengan KNN pada Klasifikasi *Genre*

# **Kesimpulan dan Saran**

## **Kesimpulan**

Dari penelitian yang telah dilakukan, terdapat 2 kesimpulan yang dapat diambil. Kesimpulan tersebut yaitu:

1. Pengaruh atribut lagu yang digunakan terhadap kinerja untuk mengklasifikasi lagu menggunakan KNN dan MKNN, diantaranya dari data lagu *genre EDM,* *Reggae*, *Soul* dan *Funk* berdasarkan variabelyang digunakantrek nya lebih cocok menari, musik *Funk* memiliki emosi yang bervariasi sementara musik *Blues* cenderung ke emosi negatif, musik rock dan metal lebih energik, musik metal memiliki tempo cepat, musik *Classical* dan *Jazz* lebih lembut dan menenangkan, musik *Hip Hop* trek musik dan ucapan vokal nya seperti musik lagu rap, musik *Classical* dan *Jazz* treknya tidak ada ucapan vokal, musik *Gospel* trek nya lebih live, musik *Classical* dan *Jazz* treknya lebih akustik, musik *Classical* dan *Funk* memiliki durasi lagu terlama, dan semua *genre* yang digunakan pada penelitian ini menggunakan trek kunci tangga nada seperti minor dan mayor.
2. Metode yang diajukan yaitu MKNN, kurang efektif dibandingkan dengan metode KNN untuk klasifikasi genre. Keefektifan tersebut terlihat dari perbandingan *k* yang paling optimal antara KNN dan MKNN terdapat selisih akurasi sebesar 1.1% sesuai dengan yang dijelaskan pada subbab 4.2. Perbandingan *k* yang paling optimal dan PCA yang paling optimal antara KNN dan MKNN terdapat selisih akurasi sebesar 1.0% sesuai dengan yang dijelaskan pada subbab 4.2. Sedangkan metode KNN tanpa pca (dengan *k* yang paling optimal) dengan metode KNN menggunakan PCA (dengan *k* yang paling optimal dan PCA yang paling optimal) memiliki selisih akurasi sebesar 0.4%. Maka dari itu, metode KNN tanpa PCA berdasarkan pada tingkat akurasi merupakan metode terbaik pada penelitian ini.

## **Saran**

Untuk pengembangan lebih lanjut adapun saran yang dapat disampaikan antara lain:

1. Menggunakan metode lain seperti Support Vector Machine (SVM) yang bisa digunakan dalam permasalahan jumlah fitur berdimensi tinggi dan menghindari ties karena metode KNN memiliki ties (Anggoro, 2020).
2. Menggunakan metode seleksi fitur lain seperti Shrinkage atau Feature Ablation (Putra, 2020).

# **DAFTAR PUSTAKA**

Anggoro, A.D., & Kurnia, N.D. (2020). Comparison of Accuracy Level of Support Vector Machine. *International Journal of Emerging Trends in Engineering Research*, 1689-1694.

Ashrith. (2018, December 04). *Analyzing Spotify’s Top Tracks Of 2017 Using Data Visualization*. Retrieved from Toward Data Science: https://towardsdatascience.com/what-makes-a-song-likeable-dbfdb7abe404

Dananjaya, D. W. (2019). Decision Support System for Classification of Early Childhood Diseases Using Principal Component Analysis and K-Nearest Neighbors Classifier. *Journal of Information Systems Engineering and Business Intelligence, Vol 5, No.1*, 13-22.

Giri, G.A.V.M. (2018). Klasifikasi Musik Berdasarkan Genre dengan Metode K-Nearest Neighbor. *Jurnal Ilmu Komputer, VOL. XI No. 2*, 103-108.

Iriansyah, F.Y. (2018, November 13). *[Update] Apa Beda Spotify dengan Apple Music, Joox, dan Deezer?* Retrieved from Techinasia: https://id.techinasia.com/komparasi-layanan-streaming-spotify

Jolliffe, I. (2002). *Principal Component Analysis, Second Edition.* New York: Springer-Verlag.

Nomleni, P. (2015). Sentiment Analysis Menggunakan Support Vector Machine (SVM). Surabaya: Program Pasca Sarjana Bidang Keahlian Telematika (CIO) Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember.

Okfalisa., Mustakim., Gazalba, I. & Reza, N.G.I. (2017). Comparative Analysis of K-Nearest Neighbor and Modified K-Nearest Neighbor Algorithm for Data Classification. *International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 294-298.

Parsian, M. (2015). *Data Algorithms: Recipes for Scaling up with Hadoop and Spark.* California: O'Reilly Media, Inc.

Parvin, H., Alizadeh, H. & Minati, B. (2010). A Modification on K-Nearest Neighbor Classifier. *Global Journal of Computer Science and Technology*, 37-41.

Putra, J.W.G. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning.* Jepang.

Ravi, M.R., Indriati., & Adinugroho, S. (2019). Implementasi Algoritme Modified K-Nearest Neighbor (MKNN) Untuk Mengidentifikasi Penyakit Gigi Dan Mulut. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol 3*, 2596-2602.

Redjeki, S. (2013). Perbandingan Algoritma Backpropagation dan K-Nearest Neighbor (K-NN) untuk Identifikasi Penyakit. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta*, 1-5.

Setiawan, E. (2020, Mei 12). *Kamus Besar Bahasa Indonesia (KBBI)*. Retrieved from https://kbbi.web.id: https://kbbi.web.id/musik

Spotify. (2020, June 12). *Perusahaan Media Spotify*. Retrieved from https://www.spotify.com/: https://newsroom.spotify.com/company-info/ & https://www.spotify.com/us/about-us/contact/

Supriyadi. (2018). *Analisis Klasifikasi Genre Musik Pop dan Klasik pada Layanan Streaming Musik Spotify Menggunakan Artificial Neural Network (ANN) (Studi Kasus: Lagu dengan Genre Musik Pop dan Klasik di Layanan Streaming Musik Spotify) .* Yogyakarta: Universitas Islam Indonesia.

Wibowo, A. (2017). *Binus University Graduate Program*. Retrieved February 8, 2020, from https://mti.binus.ac.id/2017/11/24/10-fold-cross-validation/