

**KLASIFIKASI MUSIK BERDASARKAN GENRE PADA  
LAYANAN STREAMING MUSIK SPOTIFY MENGGUNAKAN  
ALGORITMA *K-NEAREST NEIGHBOR* DAN *MODIFIED  
K-NEAREST NEIGHBOR***

**SKRIPSI**



**I MADE TANGKAS WAHYU KENCANA YUDA**

**NIM. 1608561031**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**JIMBARAN**

**2020**

## **LEMBAR PENGESAHAN TUGAS AKHIR**

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

Kompetensi : Music Information Retrieval (MIR)

Nama : I Made Tangkas Wahyu Kencana Yuda

NIM : 1608561031

Tanggal Seminar : 18 Mei 2020

Disetujui oleh:

Pembimbing I

Pengaji I

I Wayan Supriana, S.Si., M.Cs.

Luh Arida Ayu Rahning Putri, S.Kom., M.Cs.

NIP. 1984082920181113001

NIP. 198209182008122002

Pembimbing II

Pengaji II

I Gede Santi Astawa, S.T., M.Cs.

I Komang Ari Mogi, S.Kom., M.Kom.

NIP. 198012062006041003

NIP. 198409242008011007

Pengaji III

Dra. Luh Gede Astuti, M.Kom.

NIP. 196401141994022001

Mengetahui,

Koordinator Program Studi Informatika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Udayana

Dr. Ir. I Ketut Gede Suhartana., S.Kom., M.Kom.

NIP. 197201102008121001

Judul	: Klasifikasi Musik Berdasarkan <i>Genre</i> pada Layanan Musik Streaming Musik Spotify menggunakan Algoritma <i>K–Nearest Neighbor</i> dan <i>Modified K–Nearest Neighbor</i>
Nama	: I Made Tangkas Wahyu Kencana Yuda (NIM: 1608561031)
Pembimbing	: 1. Bapak I Wayan Supriana, S.Si., M.Cs. 2. Bapak I Gede Santi Astawa, S.T., M.Cs.

## **ABSTRAK**

Dalam perkembangan teknologi era modern saat ini, musik dapat didengarkan melalui berbagai macam media, salah satunya media *streaming music*. Pemberian label *genre* secara manual menyebabkan ketidakefektifan pada pertumbuhan jumlah musik pada dunia digital yang sangat pesat. Penerapan algoritma kecerdasan buatan dapat di gunakan untuk pemberian label *genre* secara otomatis, oleh karena itu dilakukan klasifikasi musik berdasarkan *genre* dengan menggunakan fitur–fitur pada atribut lagu.

Beberapa penelitian yang sudah dilakukan terdapat beragam metode yang sering digunakan untuk pengklasifikasian pada *genre*, satu diantaranya adalah KNN. Dalam penelitian ini dilakukan untuk mengetahui perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan 12 fitur audio yaitu *danceability*, *valence*, *energy*, *tempo*, *loudness*, *speechiness*, *instrumentalness*, *liveness*, *acousticness*, *key*, *mode*, dan *duration* pada 14 *genre* musik yaitu *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul*, serta untuk mengetahui perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan seleksi fitur PCA.

Dari penelitian yang dilakukan dihasilkan bahwa MKNN kurang efektif dibandingkan KNN. Pada pengujian pertama, perbandingan *k* yang paling optimal KNN pada nilai *k* = 9 mendapatkan akurasi 60.7%, sedangkan MKNN pada nilai *k* = 9 mendapatkan akurasi 59.6% dengan selisih akurasi kedua metode sebesar 1.1%. Sedangkan pada pengujian kedua menggunakan seleksi fitur PCA, perbandingan *k* yang paling optimal dan PCA yang paling optimal dari akurasi PCA yang tertinggi pada PCA 11 fitur, PCA–KNN pada nilai *k* = 9 mendapatkan akurasi 60.3%, sedangkan PCA–MKNN pada nilai *k* = 9 mendapatkan akurasi 59.3% dengan selisih akurasi kedua metode sebesar 1.0%. Sedangkan metode KNN tanpa PCA pada perbandingan *k* optimal dengan metode KNN menggunakan PCA pada perbandingan *k* optimal dan PCA optimal memiliki selisih akurasi sebesar 0.4%. Maka dari itu, metode KNN tanpa PCA berdasarkan pada tingkat akurasi merupakan metode terbaik pada penelitian ini.

**Kata Kunci : Klasifikasi Musik, *K–Nearest Neighbor*, *Modified K–Nearest Neighbor***

Title	: <i>Music Classification Based on Genre on the Spotify Music Streaming Music Service using the K–Nearest Neighbor and Modified K–Nearest Neighbor Algorithms</i>
Name	: I Made Tangkas Wahyu Kencana Yuda (NIM: 1608561031)
Supervisor	: 1. Bapak I Wayan Supriana, S.Si., M.Cs. 2. Bapak I Gede Santi Astawa, S.T., M.Cs.

## ***ABSTRACT***

*In today's technological developments, music can be heard through various media, one of which is music streaming media. The manual labeling of genres is ineffective in the rapidly growing number of music in the digital world. The application of artificial intelligence can be used to present genre labels automatically, because it is done classifying music by genre using the features in the song attributes.*

*Several studies that have been carried out there are various methods that are often used to classify genres, one of which is KNN. This research was conducted to see the comparison of the accuracy produced by the KNN and MKNN algorithms in music classification based on genre with 12 audio features, namely danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, acousticness, key, mode, and duration at 14. music genres are Blues, Classical, Country, EDM, Funk, Gospel, Hip Hop, Jazz, Metal, Pop, R&B, Reggae, Rock, and Soul, as well as for the accurate accuracy produced by KNN and MKNN in music classification by genre with PCA feature selection.*

*From the research conducted, it was found that MKNN was less effective than KNN. In the first examiner, the most optimal k ratio of KNN at  $k = 9$  got an accuracy of 60.7%, while MKNN at  $k = 9$  got a value of 59.6% with the difference between the two methods of 1.1%. Whereas in the second using PCA feature selection, the most optimal k ratio and the most optimal PCA from the PCA test were the highest on PCA 11 features, PCA - KNN at  $k = 9$  got a value of 60.3%, while PCA - MKNN at  $k = 9$  gets an accuracy of 59.3% with the difference in accuracy between the two methods 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 had a difference of 0.4%. Therefore, the KNN method without PCA based on the level of accuracy is the best method in this study.*

***Keywords : Music Classification, K–Nearest Neighbor, Modified K–Nearest Neighbor***

## KATA PENGANTAR

Penelitian dengan judul “Klasifikasi Musik Berdasarkan *Genre* pada Layanan Musik *Streaming* Musik Spotify menggunakan Algoritma *K–Nearest Neighbor* dan *Modified K–Nearest Neighbor*” ini disusun dalam rangkaian kegiatan Tugas Akhir di Jurusan Ilmu Komputer, Program Studi Informatika, FMIPA UNUD. Sehubungan dengan telah terselesaikannya penelitian ini, maka diucapkan terima kasih dan penghargaan kepada berbagai pihak yang telah membantu pengusul, antara lain:

1. Bapak I Wayan Supriana, S.Si., M.Cs. sebagai calon Pembimbing I yang telah bersedia mengkritisi, memeriksa dan menyempurnakan penelitian ini.
2. Bapak I Gede Santi Astawa, S.T., M.Cs. sebagai calon Pembimbing II yang telah bersedia mengkritisi, memeriksa dan menyempurnakan penelitian ini.
3. Bapak-bapak dan Ibu-ibu dosen pengajar di Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana yang telah meluangkan waktu untuk memberikan saran dan masukan dalam menyempurnakan tugas akhir ini.
4. Kawan-kawan di Jurusan Ilmu Komputer Fakultas MIPA Universitas Udayana yang telah memberikan dukungan dalam proses penyelesaian tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih mengandung kelemahan dan kekurangan. Memperhatikan hal ini, maka adanya masukan dan saran-saran yang membangun dari pembaca sangatlah diharapkan demi peningkatan kualitas tugas akhir ini.

Bukit Jimbaran, 17 Februari 2021

Penyusun

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN TUGAS AKHIR .....	ii
ABSTRAK.....	iii
<i>ABSTRACT</i> .....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL .....	viii
DAFTAR GAMBAR.....	ix
BAB I PENDAHULUAN.....	13
1.1.    Latar Belakang .....	13
1.2.    Rumusan Masalah.....	16
1.3.    Tujuan Penelitian .....	16
1.4.    Batasan Masalah.....	16
1.5.    Manfaat Penelitian .....	17
BAB II TINJAUAN PUSTAKA .....	18
2.1.    Tinjauan Empiris .....	18
2.2.    Tinjauan Teoritis .....	21
2.2.1.    Lagu dan Musik .....	21
2.2.2.    Spotify .....	27
2.2.3.    Identifikasi Fitur Audio Spotify .....	29
2.2.4.    Transformasi Data .....	31
2.2.5.    Klasifikasi .....	32
2.2.6. <i>Principal Component Analysis (PCA)</i> .....	33
2.2.7. <i>K–Fold Cross–Validation</i> .....	34
2.2.8. <i>Confusion Matrix</i> .....	34
2.2.9. <i>K–Nearest Neighbor</i> .....	35
2.2.10. <i>Modified K–Nearest Neighbor</i> .....	36

<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>39</b>
3.1.    Pengumpulan Data .....	39
3.2.    Variabel dan Definisi Variabel .....	39
3.3.    Alur Penelitian Sistem Klasifikasi <i>Genre Musik</i> .....	41
3.4.    Proses KNN ( <i>K–Nearest Neighbor</i> ).....	43
3.5.    Proses MKNN ( <i>Modified K–Nearest Neighbor</i> ).....	44
3.6.    Tahap Pengujian dan evaluasi .....	46
3.6.1.    Pengujian Pertama .....	46
3.6.2.    Pengujian Kedua.....	46
3.6.3.    Evaluasi .....	47
3.7.    Implementasi Sistem .....	48
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>49</b>
4.1.    Pengumpulan Data.....	49
4.2.    Inisialisasi Data Spotify .....	54
4.3.    Analisis Sebaran Data .....	55
4.4.    Transformasi Data .....	61
4.5.    Implementasi KNN dan MKNN .....	64
4.3.1.    KNN .....	64
4.3.2.    MKNN .....	65
4.6.    Implementasi Sistem .....	66
4.7.    Hasil Pengujian.....	67
4.5.1.    Pengujian Pertama .....	68
4.5.2.    Pengujian Kedua.....	88
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>102</b>
5.1.    Kesimpulan .....	102
5.2.    Saran.....	103
<b>DAFTAR PUSTAKA.....</b>	<b>104</b>
<b>LAMPIRAN.....</b>	<b>107</b>

## DAFTAR TABEL

Tabel 2.1. <i>Confusion Matrix</i> .....	35
Tabel 3.1. Variabel Fitur Pengguna .....	39
Tabel 3.2. Variabel Fitur Audio.....	40
Tabel 4.1. Data <i>Playlist</i> Musik .....	55
Tabel 4.2. Data Variabel Fitur sebelum ditransformasi .....	61
Tabel 4.3. Data Variabel Fitur setelah ditransformasi.....	62
Tabel 4.4. Data Lagu 14 <i>Genre</i> setelah ditransformasi .....	63
Tabel 4.5. <i>Shuffle</i> Data.....	68
Tabel 4.6. Hasil Evaluasi <i>Confusion Matrix</i> pada Klasifikasi <i>Genre</i> Lagu menggunakan KNN saat nilai $k = 9$ di $k$ - <i>fold</i> iterasi ke-1.....	72
Tabel 4.7. Hasil Evaluasi <i>Confusion Matrix</i> pada Klasifikasi <i>Genre</i> Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ - <i>fold</i> iterasi ke-1 .....	74
Tabel 4.8. Nilai EigenValue dan Persentase tiap Fitur .....	88

## DAFTAR GAMBAR

Gambar 1.1. Keunggulan Spotify (Iriansyah, 2018) .....	13
Gambar 3.1. Diagram Alir Sistem Klasifikasi <i>Genre</i> Musik .....	41
Gambar 3.2. Diagram Alir Proses KNN.....	43
Gambar 3.3. Diagram Alir Proses MKNN .....	44
Gambar 3.4. Lanjutan Diagram Alir Proses MKNN .....	45
Gambar 3.5. 10– <i>Fold Cross–Validation</i> .....	47
Gambar 4.1. <i>Playlist Genre Blues</i> .....	49
Gambar 4.2. <i>Playlist Genre Classical</i> .....	49
Gambar 4.3. <i>Playlist Genre Country</i> .....	50
Gambar 4.4. <i>Playlist Genre EDM</i> .....	50
Gambar 4.5. <i>Playlist Genre Funk</i> .....	50
Gambar 4.6. <i>Playlist Genre Gospel</i> .....	51
Gambar 4.7. <i>Playlist Genre Hip Hop</i> .....	51
Gambar 4.8. <i>Playlist Genre Jazz</i> .....	51
Gambar 4.9. <i>Playlist Genre Metal</i> .....	52
Gambar 4.10. <i>Playlist Genre Pop</i> .....	52
Gambar 4.11. <i>Playlist Genre R&amp;B</i> .....	52
Gambar 4.12. <i>Playlist Genre Reggae</i> .....	53
Gambar 4.13. <i>Playlist Genre Rock</i> .....	53
Gambar 4.14. <i>Playlist Genre Soul</i> .....	53
Gambar 4.15. Plot sebaran data <i>Danceability</i> .....	56
Gambar 4.16. Plot sebaran data <i>Valence</i> .....	56
Gambar 4.17. Plot sebaran data <i>Energy</i> .....	56
Gambar 4.18. Plot sebaran data <i>Tempo</i> .....	57
Gambar 4.19. Plot sebaran data <i>Loudness</i> .....	57
Gambar 4.20. Plot sebaran data <i>Speechiness</i> .....	58
Gambar 4.21. Plot sebaran data <i>Instrumentalness</i> .....	58
Gambar 4.22. Plot sebaran data <i>Liveness</i> .....	58
Gambar 4.23. Plot sebaran data <i>Accousticness</i> .....	59
Gambar 4.24. Plot sebaran data <i>Key</i> .....	59

Gambar 4.25. Plot sebaran data <i>Mode</i> .....	60
Gambar 4.26. Plot sebaran data <i>Duration</i> .....	60
Gambar 4.27. Sebaran data <i>Key</i> setelah ditransformasi.....	62
Gambar 4.28. Sebaran data <i>Loudness</i> setelah ditransformasi .....	63
Gambar 4.29. Sebaran data <i>Tempo</i> setelah ditransformasi .....	63
Gambar 4.30. Sebaran data <i>Duration</i> setelah ditransformasi.....	63
Gambar 4.31. Sebelum Normalisasi dengan sebaran data .....	66
Gambar 4.32. Sesudah Normalisasi dengan sebaran data.....	67
Gambar 4.33. Akurasi MKNN .....	70
Gambar 4.34. Perbandingan Akurasi MKNN dengan KNN pada Klasifikasi <i>Genre</i> .....	71
Gambar 4.35. Perbandingan Akurasi 10– <i>Fold Cross–Validation</i> MKNN dengan KNN pada $k = 9$ .....	72
Gambar 4.36. Hasil Akurasi Klasifikasi <i>Genre Lagu</i> menggunakan KNN pada <i>Confusion Matrix</i> saat nilai $k = 9$ di $k$ - <i>fold</i> iterasi ke–1 .....	73
Gambar 4.37. Hasil Akurasi Klasifikasi <i>Genre Lagu</i> menggunakan MKNN pada <i>Confusion Matrix</i> saat nilai $k = 9$ di $k$ - <i>fold</i> iterasi ke–1 .....	75
Gambar 4.38. Perhitungan <i>euclidean distance</i> KNN sejumlah $k = 9$ pada data <i>testing</i> ke-0 .....	76
Gambar 4.39. Kelas Data Hasil Klasifikasi KNN pada data <i>testing</i> ke-0 yang terpilih benar .....	76
Gambar 4.40. Perhitungan <i>euclidean distance</i> KNN sejumlah $k = 9$ pada data <i>testing</i> ke-4 .....	77
Gambar 4.41. Kelas Data Hasil Klasifikasi KNN pada data <i>testing</i> ke-4 yang terpilih salah.....	78
Gambar 4.42. Perhitungan <i>euclidean distance</i> MKNN sejumlah $k = 9$ pada data <i>testing</i> ke-0.....	78
Gambar 4.43. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-0 ....	79
Gambar 4.44. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-1 ....	80
Gambar 4.45. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-2 ....	80
Gambar 4.46. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-3 ....	80
Gambar 4.47. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-4 ....	81

Gambar 4.48. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-5 ....	81
Gambar 4.49. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-6 ....	81
Gambar 4.50. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-7 ....	82
Gambar 4.51. Menghitung validasi dan bobot data <i>testing</i> ke-0 tetangga ke-8 ....	82
Gambar 4.52. Kelas Data Hasil Klasifikasi MKNN pada data <i>testing</i> ke-0 yang terpilih benar .....	82
Gambar 4.53. Perhitungan <i>euclidean distance</i> MKNN sejumlah $k = 9$ pada data <i>testing</i> ke-4.....	83
Gambar 4.54. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-0 ....	84
Gambar 4.55. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-1 ....	84
Gambar 4.56. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-2 ....	85
Gambar 4.57. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-3 ....	85
Gambar 4.58. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-4 ....	85
Gambar 4.59. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-5 ....	86
Gambar 4.60. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-6 ....	86
Gambar 4.61. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-7 ....	87
Gambar 4.62. Menghitung validasi dan bobot data <i>testing</i> ke-4 tetangga ke-8 ....	87
Gambar 4.63. Kelas Data Hasil Klasifikasi MKNN pada data <i>testing</i> ke-4 yang terpilih salah.....	87
Gambar 4.64. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 1 fitur baru .....	90
Gambar 4.65. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 2 fitur baru .....	91
Gambar 4.66. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 3 fitur baru .....	92
Gambar 4.67. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 4 fitur baru .....	93
Gambar 4.68. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 5 fitur baru .....	94
Gambar 4.69. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 6 fitur baru .....	95

Gambar 4.70. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 7 fitur baru .....	96
Gambar 4.71. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 8 fitur baru .....	98
Gambar 4.72. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 9 fitur baru .....	99
Gambar 4.73. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 10 fitur baru .....	100
Gambar 4.74. Perbandingan Akurasi PCA–MKNN dengan PCA–KNN pada 11 fitur baru .....	101

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

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, seperti dijelaskan pada Gambar 1.1 (Iriansyah, 2018). Pengguna Spotify dapat mengakses musik berdasarkan *artist*, *album*, *genre*, *playlist*, *podcasts*, atau label rekaman (Spotify, 2020).

Perbandingan Layanan Streaming Musik				
Mana Layanan	Spotify	Apple Music	JOOX	DEEZER
Asal Negara	Sweden	Amerika	Prancis	Cina
Jumlah Pengguna	99+ juta pengguna (27+ Juta bayar) *Okt 18	90+ juta pengguna (berbayar) *Mai 18	14+ juta pengguna *Nov 18	Tidak ada data
Jumlah Lagu	35+ juta lagu *Okt 18	50+ juta lagu *Nov 18	53+ juta lagu *Nov 18	Tidak ada data
Kualitas Audio	Up to 320kbps	Up to 256kbps	Up to 141kbps (FLAC)	Up to 320kbps
Video	Ada video	Ada video	Tidak ada video	Ada video
Movie Offline	Ada	Ada	Ada	Ada
Versi Gratis	Ada	Tidak ada	Ada	Ada
Periode Trial	Ada 1 bulan	Ada 3 bulan	Ada 1 bulan	Ada 1 bulan
Family Plan	Ada	Ada	Tidak Ada	Tidak Ada
Urin Lagu	Lirik hanya di beberapa lagu	Lirik hanya di beberapa lagu	Lirik hanya di beberapa lagu	Lirik hanya di beberapa lagu

Gambar 1.1. Keunggulan Spotify (Iriansyah, 2018)

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 diorga-

nisasi berdasarkan *genre* musik memberikan kemudahan pada pendengar musik untuk mencari musik–musik sejenis yang sesuai dengan referensinya. Pemberian label genre secara manual menyebabkan ketidak efektifan pada pertumbuhan jumlah musik pada dunia digital yang sangat pesat. Penerapan algoritma kecerdasan buatan dapat di gunakan untuk pemberian label genre secara otomatis, hal tersebut dapat mengklasifikasi musik berdasarkan genre dengan menggunakan fitur–fitur pada atribut lagu.

Pemberian label *genre* musik yang ada di Spotify cukup banyak, 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*, sehingga total *playlist* yang ada di Spotify berjumlah ribuan. Pembagian musik berdasarkan *genre* ini tentunya bertujuan untuk mempermudah pengguna Spotify memilih lagu kesukaannya sesuai *genre* maupun *playlist*.

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 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*) dengan menggunakan Spotify API. 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%.

Selanjutnya Okfalisa, dkk (2017) pada penelitiannya melakukan perbandingan akurasi metode klasifikasi *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN). Analisis komparatif didasarkan pada keakuratan kedua algoritma. 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% .

Jika penelitian sebelumnya, hanya membahas penggunaan Spotify API dengan 11 fitur audio dengan metode KNN serta membahas metode *k-fold cross-validation* dan *confusion matrix*, maka pada penelitian ini penulis ingin menggunakan 12 fitur audio antara lain *danceability*, *valence*, *energy*, *tempo*, *loudness*, *speechiness*, *instrumentalness*, *liveness*, *acousticness*, *key*, *mode*, dan *duration* dengan 14 *genre* musik pada Spotify yaitu *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* dengan membandingkan kedua metode algoritma klasifikasi *K–Nearest Neighbor* (KNN) dengan metode algoritma klasifikasi *Modified K–Nearest Neighbor* (MKNN) menggunakan *k-fold cross-validation* untuk membagi sampel data secara acak dan mengelompokkan data tersebut sebanyak nilai *10-fold* dan mengukur seberapa tepat klasifikasi terhadap kelas menggunakan *confusion matrix* untuk akurasi pada tiap kelas *genre*. 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 klasifikasi *Modified K–Nearest Neighbor* (MKNN) menggunakan *k-fold cross-validation* dan mengukur seberapa tepat

klasifikasi terhadap kelas menggunakan *confusion matrix* dengan akurasi pada tiap kelas *genre*.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan dari penelitian ini, yaitu sebagai berikut.

- a. Bagaimana perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan 12 fitur audio yaitu *danceability*, *valence*, *energy*, *tempo*, *loudness*, *speechiness*, *instrumentalness*, *liveness*, *acousticness*, *key*, *mode*, dan *duration* pada 14 *genre* musik yaitu *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul*?
- b. Bagaimana perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan seleksi fitur?

### **1.3. Tujuan Penelitian**

Berdasarkan rumusan masalah tersebut, tujuan dari penelitian ini adalah sebagai berikut.

- a. Untuk mengetahui perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan 12 fitur audio yaitu *danceability*, *valence*, *energy*, *tempo*, *loudness*, *speechiness*, *instrumentalness*, *liveness*, *acousticness*, *key*, *mode*, dan *duration* pada 14 *genre* musik yaitu *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul*.
- b. Untuk mengetahui perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan seleksi fitur.

### **1.4. Batasan Masalah**

Beberapa batasan masalah yang akan dijadikan acuan dalam penelitian ini adalah sebagai berikut.

- a. Data yang digunakan adalah lagu dengan 14 *genre* musik yang ada di layanan *streaming music* Spotify yaitu *Blues, Classical, Country, EDM, Funk, Gospel, Hip Hop, Jazz, Metal, Pop, R&B, Reggae, Rock, dan Soul*.
- b. Variabel fitur pada atribut lagu yang digunakan adalah 12 fitur audio, antara lain *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode, dan duration*.
- c. Perbandingan metode yang digunakan adalah metode algoritma *K–Nearest Neighbor* dan *Modified K–Nearest Neighbor*.

### **1.5. Manfaat Penelitian**

Beberapa manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut.

a. Bagi Masyarakat

Penelitian ini dapat membantu untuk mendapatkan informasi tentang perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* dengan 12 fitur audio yaitu *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode, dan duration* pada 14 *genre* musik yaitu *Blues, Classical, Country, EDM, Funk, Gospel, Hip Hop, Jazz, Metal, Pop, R&B, Reggae, Rock, dan Soul*.

b. Bagi Penulis

Penelitian ini diharapkan dapat menambah pengetahuan bagi penulis mengenai perbandingan akurasi yang dihasilkan algoritma KNN dan MKNN pada klasifikasi musik berdasarkan *genre* tanpa menggunakan seleksi fitur ataupun dengan seleksi fitur.

c. Bagi Keilmuan

Penelitian ini dapat menjadi referensi perbandingan algoritma KNN dan MKNN pada penelitian lain yang memiliki karakteristik yang sama.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bagian ini, akan dipaparkan sejumlah tinjauan empiris dan tinjauan teoritis yang akan dijadikan acuan dalam penelitian mengenai klasifikasi menggunakan metode *K–Nearest Neighbor* (KNN) dengan *Modified K–Nearest Neighbor* (MKNN).

#### **2.1. Tinjauan Empiris**

Terdapat beberapa penelitian serupa baik pendekatan metode *K–Nearest Neighbor* (KNN) dengan *Modified K–Nearest Neighbor* (MKNN), maupun lingkup penelitian yang mirip seperti penelitian ini. Sehingga dari penelitian sebelumnya dapat dijadikan tinjauan pustaka oleh penulis.

**A. 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) oleh (Supriyadi, 2018)

Dalam penelitiannya metode *Backpropagation* digunakan untuk melakukan pengklasifikasian 2 *genre* musik yaitu pop dan klasik. Data input yang digunakan adalah 7 fitur audio. Antara lain, *accousticness*, *danceability*, *energy*, *loudness*, *speechiness*, *tempo* dan *valence*, kemudian outputnya adalah *genre* musik *pop* dan *klasik*. Dari hasil mengimplementasikan ANN dengan metode *backpropagation* digunakan arsitektur jaringan saraf tiruan 7 neuron input, 1 hidden layer dengan 4 neuron dan 1 output. Hasil akurasi dari pengujian diperoleh sebesar 99,5%. Baik pada penelitian Supriyadi (2018) maupun peneltian penulis sama-sama melakukan klasifikasi musik berdasarkan *genre* pada layanan *streaming music* Spotify dengan fitur audio.

Perbedaannya, jika pada penelitian Supriyadi (2018) menggunakan 2 *genre* musik yaitu pop dan klasik dengan algoritma *Backpropagation* dan menggunakan 7 fitur audio, maka pada penelitian ini penulis menggunakan 12 *genre* musik dengan algoritma KNN dan MKNN menggunakan 12 fitur audio.

**B. Implementasi Algoritme Modified K-Nearest Neighbor (MKNN) Untuk Mengidentifikasi Penyakit Gigi Dan Mulut** oleh (Ravi, dkk, 2019)

Dalam penelitiannya, dilakukan identifikasi jenis penyakit gigi dan mulut yang ditentukan dari gejala yang dialami dengan menggunakan metode klasifikasi *Modified K-Nearest Neighbor* (MKNN) menggunakan 6 kelas data yang meliputi Pulpitis, Gingivitis, Karies Gigi, Periodontitis, Deposits, dan Nekrosis Pulpa. Pada penelitian ini membuktikan bahwa pada data *training* (data latih) sebanyak 70 dan data *testing* (data uji) 30 serta nilai  $k = 60$  dan 70, metode MKNN dapat melakukan identifikasi jenis penyakit gigi dan mulut dengan mencapai 86,6%. Pada penelitian ini juga membuktikan bahwa metode MKNN cenderung lebih tinggi akurasinya dibandingkan dengan metode KNN dimana metode MKNN memiliki tingkat akurasi 76,66% sedangkan KNN 43,33%. Baik pada penelitian Ravi, dkk (2018) maupun peneltian penulis sama-sama menggunakan metode klasifikasi *Modified K-Nearest Neighbor* (MKNN).

Perbedaannya, jika pada penelitian Ravi, dkk (2019) melakukan klasifikasi identifikasi jenis penyakit gigi dan mulut, maka pada penelitian ini penulis melakukan klasifikasi musik pada layanan *streaming* Spotify.

**C. A Modification on K-Nearest Neighbor Classifier** oleh (Parvin, Alizadeh & Minati, 2010)

Pada KNN, terdapat permasalahan outlier untuk perhitungan bobot. Sehingga ditambahkan validasi dari setiap data latih. Validasi tersebut dilakukan dengan menghitung kesamaan kelas data antar data latih yang berdekatan. Permasalahan yang diajukan untuk melakukan klasifikasi dari kelas data Monk 1, Monk 2, Monk 3, Isodata, Wine, Iris, Balance-sc, Bupa & SHeart. Pengujian dilakukan sebanyak 500 data uji dengan menggunakan bilangan ganjil kisaran  $k$  [3-15], yang kemudian dirata-ratakan. Adapun hasil dari penelitian yang dilakukan sebagai berikut:

Dari keseluruhan pengujian, seluruh 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. Dengan nilai akurasi yang tinggi, maka klasifikasi data uji setiap *genre* pada penelitian yang diajukkan menggunakan metode MKNN.

Penelitian Parvin, dkk (2010) menjadi latar belakang penulis dalam memilih metode *Modified K–Nearest Neighbor* (MKNN) sebagai metode dalam melakukan klasifikasi musik berdasarkan *genre* pada layanan *streaming music* Spotify.

**D. Decision Support System for Classification of Early Childhood Diseases Using Principal Component Analysis and K-Nearest Neighbors Classifier**  
oleh (Dananjaya, werdiningsih & semiati, 2019)

Dalam penelitiannya melakukan penelitian yang bertujuan untuk mengimplementasikan metode *Principal Component Analysis* (PCA) dan *K–Nearest Neighbor* (K-NN) dalam klasifikasi penyakit anak usia dini. Analisis data dilakukan dengan PCA untuk mendapatkan variabel yang memiliki pengaruh besar terhadap klasifikasi penyakit anak usia dini. PCA melakukan reduksi fitur dengan melihat korelasi antar variabel dan mengeliminasi variabel yang memiliki pengaruh kecil terhadap klasifikasi. Selanjutnya data penyakit anak usia dini diklasifikasikan menggunakan metode *K–Nearest Neighbor Classifier*. Hasil evaluasi sistem dengan menggunakan 150 data uji dengan 14 variabel fitur baru menunjukkan bahwa sistem klasifikasi dengan menerapkan PCA dan KNN Classifier memiliki nilai akurasi sebesar 86%, sedangkan dengan 18 variabel fitur baru tanpa menerapkan PCA memiliki nilai akurasi sebesar 72%.

Perbedaannya, jika pada penelitian Dananjaya, dkk (2019) melakukan reduksi variabel fitur pada data penyakit anak usia dini, maka pada penelitian ini penulis melakukan reduksi variabel fitur pada data klasifikasi musik berdasarkan *genre* pada layanan *streaming music* Spotify.

## 2.2. Tinjauan Teoritis

### 2.2.1. Lagu dan Musik

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).

Lagu adalah ragam suara yang berirama seperti dalam bercakap, bernyanyi, membaca dan lain sebagainya. Selain itu lagu sering dikenal dengan sebuah syair maupun lirik yang memiliki irama, sebuah lagu biasanya diiringi dengan alat musik untuk memperoleh musik yang mengandung suara berirama. Secara umum lagu memang tidak lepas dengan musik. Selanjutnya pengertian musik merupakan sesuatu yang menghasilkan sebuah karya seni, seperti berupa musik berbentuk lagu atau komposisi yang mengungkapkan isi pikiran serta perasaan seorang pencipta lagu lewat unsur-unsur pokok musik, yaitu irama, melodi, harmoni, tempo, birama, timbre, tangga nada, dinamika serta ekspresi sebagai satu kesatuan susunan lagu (Jamalus,1988).

Dari beberapa banyak lagu yang ada saat ini, terbagi lagi menjadi beberapa *genre* atau jenis musik yang sesuai dengan lagu tersebut. *Genre* merupakan pengelompokan musik yang sesuai dengan kemiripan antara satu sama lainnya. Pada umumnya sebuah *genre* musik dapat didefinisikan berdasarkan teknik musik, gaya, konteks, maupun temanya. *Genre* secara umum berarti tipe atau kelas dari musik yang kita dengar. *Genre* musik memberikan ekspektasi bagaimana suara musik tersebut, berapa lama musik tersebut, dan bagaimana pendengar harus berperilaku. Pada era Mozart, terdapat lima *genre* utama yaitu *symphony*, *string quartet*, *sonata*, *concerto*, dan opera (Giri, 2018). Pada era musik modern, musik dapat dibagi menjadi berbagai *genre* seperti *Blues*, *Classical*, *Country*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *Reggae*, *Rock*, *EDM*, *Funk*, *R&B*, dan *Soul*. Deskripsi dari *genre–genre* tersebut adalah sebagai berikut:

#### 1. Musik *Blues*

Musik *Blues* merupakan sebuah aliran musik yang berasal dari Amerika Serikat. *Blues* merupakan perkembangan dari musik-musik spiritual serta

pujian-pujian yang berasal dari komunitas budak-budak Afrika di AS. Musik ini hanya populer dikalangan orang Amerika pada sebelum abad ke-20. Adapun ciri khas musik ini yaitu pola irama yang sering digunakan, terkesan sedih dan permainan gitarnya selalu penuh dengan improvisasi (Sitinjak, 2019).

## 2. Musik *Classical*

Musik *classical* atau klasik merupakan kumpulan dari beberapa genre musik pada era Mozart yang berbentuk seperti *symphony*, *string quartet*, *sonata*, *concerto*, dan opera. Pada abad ke-9 hingga abad ke-21 musik *classical* biasanya dimainkan dengan menggunakan alat musik tertentu seperti biola serta biasanya merujuk pada musik klasik Eropa. Adapun ciri khas musik klasik seperti penggunaan dinamika *crescendo* dan *decrescendo* serta ciri lainnya seperti perubahan *tempo* dengan menggunakan gunakan *accelerando* (semakin cepat) dan *ritardando* (semakin lambat) (Supriyadi, 2018).

## 3. Musik *Country*

Musik *Country* adalah musik campuran yang berasal dari unsur – unsur musik Amerika Serikat Bagian Selatan dan Pegunungan Appalachia. Musik ini berakar dari lagu rakyat Amerika Utara, musik kelt, musik *gospel*, dan berkembang sejak tahun 1920-an di Amerika bagian selatan, saat itu musik *country* dimainkan memakai gitar, bass, dan steel guitar. Istilah musik *country* mulai dipergunakan pada tahun 1940-an untuk menggantikan istilah musik *hillbilly* yang berkesan merendahkan. Pada tahun 1970-an, istilah musik *country* telah menjadi istilah populer, serta pada abad 19 musik ini mulai menyebar luas pada orang – orang yang berimigrasi seperti dari Irlandia, Inggris, Jerman, Spanyol dan Italia menuju Texas, selanjutnya ditempat tersebut mereka bertemu dengan orang Meksiko-Amerika. Karena Texas mempunyai padang rumput yang subur sehingga para imigran tersebut banyak yang menjadi penggembala sapi, hingga akhirnya membuat Texas dikenal sebagai kota Cowboy. Seiring berjalannya waktu musik *Country* mulai diterima di Amerika Serikat hingga akhirnya menyebar ke seluruh dunia. Istilah lain genre musik ini adalah *country and western*, namun istilah tersebut semakin

jarang dipakai kecuali di Britania Raya dan Irlandia. Contoh orang yang menggunakan *Genre Country*: Taylor Swift (Sitinjak, 2019).

#### 4. Musik *Gospel*

Musik *Gospel* merupakan salah satu jenis musik religius yang pertama kali dikenalkan oleh gereja-gereja Afro-Amerika pada tahun 1930-an dan dibawakan oleh kalangan kulit hitam. Dalam perkembangannya jenis musik ini terpecah menjadi dua aliran yaitu, *Gospel* kulit hitam dan *Gospel* kulit putih. Hal ini terjadi karena adanya diskriminasi ras pada saat itu di Amerika.

*Genre Gospel* mempunyai ciri khas seperti vokal yang dominan dengan harmoni yang kuat dan liriknya yang religius, khususnya Kristen. *Genre Gospel* juga identik dengan paduan suara, koor dan choir. Lambat laun musisi *gospel* mencoba untuk memperkenalkan jenis musik ini ke kancah yang lebih luas, salah satunya Mahalia Jackson membawakan *Genre Gospel* setelah dikombinasikan dengan genre musik lain. Selanjutnya diikuti oleh Golden Gate Quartet dan Clara Ward yang lebih berani membawakan tembang-tebang gereja ini di club malam pada performnya. Hingga sampai saat ini musik *Gospel* sudah lagi tidak ada perbedaan yang kentara dengan musik kontemporer (Sitinjak, 2019).

#### 5. Musik *Hip Hop*

Musik *Hip Hop* merupakan salah satu *genre* musik yang terdiri dari perpaduan *rapping*, *DJing*, *Breakdance* dan *Graffiti*. *Hip Hop* dimulai sekitar tahun 1970-an yang dipelopori oleh kalangan Afro-Amerika dan Amerika Latin. Awal mulanya *Hip Hop* hanyalah musik dari *Disk Jockey* yang membuat variasi dengan memutar disk sehingga menciptakan bunyi yang unik. Kemudian *Rapping* masuk untuk mengisi vokal dari bunyi-bunyi tersebut yang menjadikan ciri khas dari musik *Hip Hop*, sedangkan pada koreografinya diisi dengan tarian patah-patah yang biasa disebut *breakdance*. Seperti *genre* musik yang lain, *Hip Hop* juga memiliki era dan melahirkan musisi dalam perkembangannya. Diantara musisi *Hip Hop* yang terkenal ialah Nate Dogg, Snoop Dogg and Ludacris (Sitinjak, 2019).

#### 6. Musik *Jazz*

Musik *jazz* adalah jenis musik yang tumbuh dari penggabungan musik blues, ragtime, dan musik eropa terutama musik band. Kata *Jazz* berasal dari bahasa slang (bahasa daerah pinggiran pantai barat Amerika Serikat), istilah ini pertama kali digunakan pada tahun 1915 di Chicago. Genre musik ini berasal dari masyarakat Afro–Amerika Selatan pada akhir abad ke–19 dan awal abad ke–20. Ciri khas dari *genre* ini yaitu banyak menggunakan instrumen biola, piano, saksofon, bass, drum, gitar, trombone dan terompet. Perkembangan musik *Jazz* bisa dibilang menakjubkan, dari yang awalnya dina bertransformasi ke dalam musik kelas atas. (Sitinjak, 2019).

## 7. Musik *Metal*

Musik *Metal* mengutamakan permainan gitar dan biasanya dimainkan dengan *tempo* lagu yang sangat cepat. Musik *Metal* berasal dari kata *Heavy Metal* yang terdapat pada lagu band Hard *Rock* di tahun 1960–an, *Steppenwolf*. Seiring berkembangnya musik *rock* pada tahun 1970–an, nama *Heavy Metal* kemudian dijadikan aliran tersendiri dengan merujuk pada *blues rock* dan *psychedelic rock*. *Genre* baru ini memiliki ciri seperti solo gitar yang panjang, distorsi gitar yang kuat dan ketukan yang cepat di semua instrumen alat musiknya.

Musik *Heavy metal* pada era saat ini telah melahirkan band-band besar seperti Led Zeppelin, Black Sabbath dan Deep Purple, namun elemen *blues* masih terlihat kental memengaruhi diera ini. Selanjutnya Judas Priest mengimprovisasi *genre* ini dengan menghilangkan unsur *blues* sehingga hanya mengandalkan distorsi, beat yang lebih cepat dan harmoni.

Seiring perkembangan musik yang terjadi dalam sejarah, jenis musik ini pada akhirnya menjadikannya terbagi menjadi beberapa sub–*genre* diantaranya seperti, *Deathcore*, *Metalcore*, *Progressive Metal*, *Mathcore*, *Nu Metal* atau *Groove Metal*, *Glam Metal*, *Death Metal*, *Doom Metal* atau *Ghotic Metal*, *Black Metal*, *Trash Metal*, *Folk Metal*, *Grindcore*, *Alternative Metal* dan *Power Metal* (Sitinjak, 2019).

## 8. Musik *Pop*

Musik *pop* atau musik populer merupakan jenis musik yang sangat digemari oleh masyarakat awam pada saat ini. *Genre* ini mulai berkembang dan dikenal pada tahun 1950-an. Suatu *Genre* dinamakan populer karena memiliki daya tarik masa yang lebih besar dari pada *genre* lain. Pada masa awalnya musik pop merupakan pengiring tarian tango yang bernada minor dan melankolis. Hingga sampai tahun 1940 musik Pop menjadi salah satu jenis seni musik yang diminati oleh seluruh dunia dengan irama seperti Rhumba, Samba, Salsa, Conga, Mambo dan yang lainnya. Namun selama perkembangannya Musik Pop sering tumpang tindih dengan genre lain, karena banyak musisi Pop dimasukkan ke dalam kategori rock, hiphop, country dan lain sebagainya. Adapun ciri khas musik *pop* seperti melodi yang digunakan mudah diterapkan dengan berbagai karakter lirik, fleksibel dan mudah dipadukan dengan jenis lain, lagu mudah disenandungkan dan mudah dipahami, harmoni tidak rumit, *tempo*nya dapat divariasikan (Supriyadi, 2018).

#### 9. Musik *Reggae*

Musik Reggae berasal dari Jamaika dan mirip dengan gaya musik ska dan rock steady. Genre ini mempunyai ciri khas dengan ritme *backbeat* dan progresif kord sederhana. Walaupun unsur utama *Reggae* dari *ska* dan *rocksteady*, salah satu jenis musik ini juga mengadopsi *jazz*, *R&B* dan Jamaican mento. Amerika Serikat dan Jamaika mengakui *Reggae* sebagai aliran musik tersendiri pada tahun 1960-an. Pada awalnya musik reggae sering dianggap dengan jiwa muda, *rude boys* atau anak laki-laki kasar dan pemberontakan. Namun *genre* musik ini dapat melahirkan beberapa musisi hebat seperti Bob Marley, Prince Buster, Desmond Dekker dan Jackie Mittoo sebagai legendanya meskipun sebetulnya masih banyak lagi yang lain (Sitinjak, 2019).

#### 10. Musik *Rock*

Musik *Rock* berasal dari kombinasi dari *genre* musik *rhythm* dan *blues*, *country*, *jazz*, musik klasik termasuk musik rakyat (*folk music*). Musik ini mulai dikenal secara umum pada pertengahan tahun 1950-an. Genre ini mempunyai ciri khas yang didominasi oleh vokal, gitar, drum, keyboard maupun *synthesizer*. Musik *Rock* mirip dengan metal, mengutamakan

permainan gitar tetapi memiliki *tempo* lagu yang lebih lambat dari *genre* metal. Dalam beberapa dekade saja *genre* musik ini sudah menelurkan sub-*genre* baru seperti *Alternative Rock*, *Art Rock*, *Blues rock*, *Britpop*, *Country rock*, *Emo*, *Folk rock*, *Garage rock*, *Glam rock*, *Grunge*, *Hard rock*, *Heavy Metal*, *Indie rock*, *Jazz rock*, *New wave*, *Pop rock*, *Post britpop*, *Post grunge*, *Post punk*, *Progressive rock*, *Psychedelic rock*, *Punk rock*, *Rock and roll*, *Rockabilly*, *Soft rock*, *Southern rock* dan *Surf rock*. Dalam perkembangannya hingga tersebar ke seluruh dunia, musik *Rock* juga melahirkan beberapa band yang bisa dibilang legendaris seperti The Beatles, Dream Theater dan lainnya (Sitinjak, 2019).

### 11. Musik *Electronic Dance Music* (EDM)

Musik *Electronic* atau *Techno* atau dikenal dengan *Electronic Dance Music* (EDM) mulai ada sekitar tahun 1980 di Detroit, awal mula musik ini dibentuk sebagai bentuk tari dan musik pesta. Seperti namanya, *genre* musik ini mengedepankan ragam suara dari teknologi dalam musiknya. Sebagian besar musik *Techno* merupakan kombinasi *synthesizer*, hentakan drum dan sequencer. Biasanya musik *techno* menghiasi disetiap club malam dunia. Dan beberapa musisi yang terkenal dalam *genre* musik ini adalah Carl Cox dan Richie Hawtin (Sitinjak, 2019). Dalam beberapa dekade saja *genre* musik ini sudah menelurkan sub-*genre* baru seperti *Electro House*, *Progressive House*, *Big Room House*, *Melbourne Bounce*, *Deep House*, *Future House*, *Trance*, *Tropical House*, *Grime*, *Dubstep*, *Brostep* dan lain-lainnya.

### 12. Musik *Funk*

Musik *Funk* merupakan jenis musik yang berasal dari kolaborasi *genre rhythm & blues*, *jazz* dan *soul* musik yang dijadikan sebuah ritmik. Musik ini lahir pada tahun 1960-an ketika para musisi Afro-Amerika mencoba berimprovisasi. Musik *Funk* dapat dikenali dari bunyi gitar yang kuat, drum yang dominan, alunan nada yang terpotong singkat dan memberikan kesan dansa dan gembira ketika mendengarnya (Sitinjak, 2019).

### 13. Musik *Rhythm and Blues* (R&B)

Musik *R&B* atau *Rhythm* dan *Blues* adalah penggabungan dari *genre jazz*, *blues* dan *gospel*. Awalnya musik ini hanya ditujukan bagi pendengar kulit hitam. Nama *Rhythm* dan *Blues* dibuat pada akhir tahun 1940-an di Amerika Serikat sebagai istilah pemasaran agar tidak dianggap rasis.

Pada periode awal musik *R&B* memiliki titik fokus pada ritme boogie dengan musisi ternama seperti Bo Diddley dan Chuck Berry. Jenis musik ini berkembang selama bertahun-tahun sampai akhirnya menumbuhkan beberapa *genre* baru. Musik *R&B* kontemporer mempunyai susunan tersendiri yang berbeda dengan pendahulunya, dengan beralih titik fokus pada unsur *pop* (walaupun elemen *jazz*, *gospel* dan *blues* terkadang masih tetap muncul). Seiring berkembangnya zaman, kesan *genre R&B* sebagai musik kalangan kulit hitam semakin memudar dengan menyebarluasnya musik ini ke seluruh dunia (Sitinjak, 2019).

#### 14. Musik *Soul*

Musik *Soul* merupakan salah satu *genre* musik populer yang berasal dari Amerika Serikat. Genre musik ini lahir dan berkembang pada akhir 1950 dan awal 1960. Jenis musik ini menggabungkan dua aliran musik yang telah lahir sebelumnya, yakni *Rhythm and blues* serta musik *gospel* Afrika Amerika. Menurut *Rock and Roll Hall of Fame*, musik *Soul* adalah jenis musik yang lahir melalui transformasi musik *gospel* dan *R&B* menjadi satu bentuk yang lebih *funky* serta irama yang *catchy*. Sehingga irama yang dihasilkan dapat membuat tubuh bergerak mengikuti alunan musik. Hal ini yang kemudian menjadi satu hal penting dalam musik *soul* (Hazliansyah, 2012).

##### 2.2.2. Spotify

Spotify adalah layanan *streaming music* asal Swedia yang sebagai penyedia hak digital manajemen yang dilindungi konten dari label rekaman dan perusahaan media. Musik yang ada pada Spotify dapat diakses atau dicari berdasarkan *artist*, album, *genre*, *playlist*, *podcasts*, atau label rekaman (Spotify, 2020). Selain itu, Spotify juga berperan sebagai penyedia hak cipta digital atas musik yang diunggah, kemudian lagu-lagu yang tersedia merupakan lagu resmi dan pengguna tentunya

dapat menikmati koleksi yang beragam tersebut secara resmi pula. Hal yang cukup menarik dari Spotify adalah ia menawarkan dua model berlangganan, yaitu gratis dan *premium*. Pengguna gratis dan premium mendapatkan kualitas layanan yang tentunya berbeda contohnya pada Spotify Premium dapat menghapus iklan, pengguna dapat mengunduh musik untuk dinikmati secara offline dan meningkatkan kualitas audio (Spotify gratis hanya 160kbit/s menjadi hingga 320kbit/s), sedangkan Spotify gratis tidak bisa menghapus iklan dan tidak bisa dinikmati secara offline (Winarso, 2016).

Sebagai layanan *streaming music*, Spotify memiliki banyak fitur yang beragam guna memberikan pelayanan yang terbaik bagi penggunanya. Berikut beberapa fitur-fitur pengguna pada atribut lagu tersebut (Supriyadi, 2018).

#### 1. Katalog musik

Beragamnya musik yang disediakan oleh suatu layanan *streaming music* tentu menjadi daya tarik tersendiri bagi penggunanya. Pada tahun 2012 saja, Spotify mengklaim mempunyai 20 juta koleksi lagu dari berbagai *genre* musik di seluruh dunia.

#### 2. Playlist

*Playlist* yang disediakan oleh aplikasi resmi Spotify sangat banyak, selain itu pengguna juga bisa membuat *playlist* musik sendiri sesuai dengan musik yang disukai, sehingga lagu-lagu yang disukai dapat diputar dengan mudah tanpa harus mencari-cari lagi.

#### 3. Radio

Spotify juga menyediakan akses ke beberapa stasiun radio populer dunia, disini pengguna dapat memilih sendiri *genre* musik yang digemari.

#### 4. Mode *Offline*

Bagi pengguna *premium*, fitur ini dapat digunakan tetapi tidak untuk pengguna gratis. Fitur ini dapat memutar musik yang sudah diunduh sebelumnya tanpa harus terhubung ke internet terlebih dahulu.

#### 5. Integrasi Media Sosial

Spotify memungkinkan akun penggunanya untuk terintegrasi dengan akun media sosial pengguna tersebut yaitu Facebook dan Twitter. Setelah

terhubung mereka dapat mengakses musik favorit dan juga dapat mengikut *playlist* dari teman–temannya.

### **2.2.3. Identifikasi Fitur Audio Spotify**

Untuk setiap lagu di platform mereka, Spotify menyediakan data untuk 13 fitur audio pada atribut lagu. Panduan pengembang API Web Spotify mendefinisikannya sebagai berikut (Ashrith, 2018):

1. *Danceability*

Menjelaskan seberapa kecocokan sebuah lagu atau trek musik untuk menari berdasarkan kombinasi elemen musik termasuk tempo, stabilitas ritme, kekuatan beat, dan keteraturan keseluruhan. Nilai *danceability* antara 0,0 sampai dengan 1,0, semakin mendekati 1,0 maka trek pada musik tersebut semakin cocok untuk menari begitu pula sebaliknya.

2. *Valence*

Mengindikasi jenis emosi. Lagu dengan *valence* tinggi terdengar lebih positif (mis. Bahagia, ceria, gembira), sedangkan trek dengan *valence* rendah terdengar lebih negatif (misal sedih, tertekan, marah). Hal ini juga diukur pada skala 0,0 sampai 1,0, dengan nilai yang lebih rendah yang mewakili emosi negatif dan nilai yang lebih tinggi mewakili emosi positif.

3. *Energy*

Merupakan ukuran intensitas dan aktivitas suatu tingkat dari kemampuan suatu musik untuk meningkatkan emosi dari pendengarnya. Maksudnya ada suatu musik yang memiliki energi meningkat dan ada juga yang menurun. Biasanya kontrol pada energi musik dapat dilakukan dengan meningkatkan jumlah instrumen, mengatur dinamika, maupun irama pada musik tersebut. Semakin energik suatu musik, maka nilai energi akan semakin besar, begitu pula sebaliknya, semakin lembut suatu musik, maka semakin kecil nilai energinya. Sebagai contoh, death metal memiliki energi tinggi, sedangkan prelude Bach mendapat skor rendah pada skala.

4. *Tempo*

*Tempo* merupakan cepat atau lambatnya sebuah lagu dan ukuran dari sebuah *tempo* adalah *beat*. *Beat* sendiri dapat diartikan sebagai ketukan dasar yang menunjukkan banyaknya ketukan dalam satuan menit *Beats Per Minute* (BPM).

#### 5. *Loudness*

Keseluruhan kenyaringan pada trek lagu dalam bentuk *decibels* (dB). *Loudness* dihitung dalam desibel dengan mengukur intensitas gelombang audio selama durasi lagu. Lagu yang lebih keras cenderung lebih energik atau agresif, sedangkan lagu yang lembut cenderung menggunakan instrumen yang lembut dan menggambarkan emosi yang lebih tenang.

#### 6. *Speechiness*

Mendeteksi keberadaan kata-kata yang diucapkan di dalam sebuah trek lagu, seperti eksklusif pidato atau rekaman (misal acara bincang-bincang, buku audio, puisi). Nilai di atas 0,66 menggambarkan trek yang mungkin seluruhnya terbuat dari kata-kata yang diucapkan. Nilai antara 0,33 dan 0,66 menggambarkan trek yang mungkin berisi musik dan ucapan, baik dalam bagian atau berlapis, termasuk kasus seperti musik rap. Nilai di bawah 0,33 kemungkinan besar mewakili musik dan trek *nonspeech*.

#### 7. *Instrumentalness*

Memprediksi apakah suatu lagu mengandung vokal atau tidak. Suara "Ooh" dan "Aah" diperlakukan sebagai instrumen dalam konteks ini. Rap atau trek kata yang diucapkan jelas "vokal". Nilai *instrumentalness* mempunyai range dari 0,0 hingga 1,0. Semakin dekat ke nilai 1,0, semakin tinggi kecenderungan musik tersebut tidak disertai dengan vokal.

#### 8. *Liveness*

Mendeteksi keberadaan *audiens* dalam rekaman. Nilai *liveness* yang lebih tinggi mewakili probabilitas yang meningkat bahwa trek dilakukan secara *live* (langsung).

#### 9. *Acousticness*

Suatu ukuran yang menunjukkan apakah trek tersebut akustik atau tidak. Ukuran kepercayaan dari 0,0 hingga 1,0 dari apakah trek akustik. Semakin mendekati nilai 1.0 pada trek tersebut menunjukkan semakin tinggi kepercayaan lintasannya akustik, begitu pula sebaliknya apabila semakin mendekati 0,0 maka semakin rendah kepercayaan lintasnya akustik.

#### 10. *Key*

Kunci keseluruhan trek yang diperkirakan. *Integer* memetakan ke *pitch* menggunakan *standard Pitch Class notation*. Misalnya 0 = C, 1 = C#/Db, 2 = D, dan seterusnya.

#### 11. *Mode*

Mengindikasikan modalitas (besar atau kecil) dari suatu trek, jenis skala dari mana konten melodinya diturunkan. Tangga nada mayor diwakili oleh 1 dan tangga nada minor adalah 0. Tangga nada mayor bersifat Menggembirakan, sedangkan tangga nada minor bersifat Menyedihkan. Tangga nada mayor bernada dasar C=do, sedangkan tangga nada minor bernada dasar A=la.

#### 12. *Duration*

Panjang durasi trek musik dalam satuan milidetik.

#### 13. *Time\_Signature*

Perkiraan tanda birama keseluruhan suatu trek. Tanda birama (meter) adalah konvensi notasi untuk menentukan berapa banyak ketukan disetiap panjang (atau ukuran lagu).

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*, *acousticness*, *key*, *mode* dan *duration* yang diambil dengan menggunakan Spotify API.

#### **2.2.4. 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).

## 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.

### 2.2.5. Klasifikasi

Klasifikasi adalah salah satu tugas dari *data mining* yang bertujuan untuk memprediksi label kategori benda yang tidak diketahui sebelumnya, dalam membedakan antara objek yang satu dengan yang lainnya berdasarkan atribut atau fitur (Mutrofin, dkk 2014).

Di dalam klasifikasi terdapat dua pekerjaan yang dilakukan, (Prasetya, 2012) yaitu sebagai berikut:

- Pembangunan model sebagai *prototype* untuk disimpan sebagai memori.
  - Penggunaan model tersebut untuk melakukan pengenalan klasifikasi atau prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

Terdapat beberapa banyak algoritma klasifikasi yang sudah dikembangkan oleh para peneliti diantaranya adalah *K–Nearest Neighbor* (KNN), *Modified K–Nearest Neighbor* (MKNN), *Decision Tree Classifier*, *Rule–Based Classifier*,

*Neural–Network, Artificial Neural Network (ANN), Support Vector Machine, dan Naive Bayes Classifier.*

#### **2.2.6. Principal Component Analysis (PCA)**

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).

$$\text{var}(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{(n-1)} \quad \dots \dots \dots \quad (2.2)$$

2. Menghitung *covariance*

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

$$\text{var}(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)} \quad \dots \dots \dots \quad (2.3)$$

3. Menghitung *eigenvalue* dan *eigenvectors*

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

$$(A - \lambda I)X = 0 \quad \dots \dots \dots \quad (2.4)$$

$$\det(A - \lambda I) = 0 \quad \dots \dots \dots \quad (2.5)$$

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

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

$$r_{yik,xi} = \frac{\hat{e}_{ik}\sqrt{\lambda_k}}{S_{ii}} \quad \dots\dots\dots (2.6)$$

Keterangan:

$A$  : Matrix n x n.

$\lambda$  : Eigenvalue.

$I$  : Identity Matrix.

$r_{yik,xi}$  : Korelasi antara variabel utama dengan komponen utama.

$\hat{e}_{ik}$  : eigenvectors.

$S_{ii}$  : Covariance Matrix.

### 2.2.7. 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 validasi atau evaluasi. 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).

### 2.2.8. 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 2.1 menunjukkan *confusion matrix* untuk *classifier*.

Tabel 2.1. *Confusion Matrix*

		Data Aktual	
		Ya	Tidak
Data Prediksi	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 kedekatan antara prediksi benar (*positif*) dengan aktual (sebenarnya) dari keseluruhan data.

### 2.2.9. 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).

## Keterangan:

*Euclidean Distance*( $x, y$ ): Jarak antara data latih dengan data uji.

$x_i$  : Atribut fitur data latih.

$y_i$  : 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).

### **2.2.10. 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 latih dengan data uji pada Persamaan(2.8). Kemudian diurutkan dan dipilih sebanyak  $k$  yang terpendek.
- Tahap kedua, dari setiap data latih terpendek ditentukan nilai validitasnya. Nilai validitas ditentukan dari data latih lainnya yang terdekat. Jumlah seluruh validitas dibagi dengan  $k$ . Rumus nilai validitas pada Persamaan (2.4).

$$\text{Validitas} = \frac{1}{k} \sum_{i=1}^k S(\text{lbl}_x, \text{lbl}_{Nix}) \dots \dots \dots \quad (2.9)$$

Keterangan:

*Validity* : Nilai validitas.

$k$  : Besaran atau jumlah titik terdekat dari individu.

$\text{lbl}_x$  : Kelas data latih.

$\text{lbl}_{Nix}$  : Kelas data latih lainnya ke– $i$  yang terdekat dengan  $\text{lbl}(x)$ .

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.

$$S(a, b) = \begin{cases} 1 & \text{jika } a = b \\ 0 & \text{jika } a \neq b \end{cases} \dots \dots \dots \quad (2.10)$$

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).

## Keterangan:

$W_{(i)}$  : Bobot data latih ke-*i*.

*Validitas(i)* : Validitas data latih ke-i.

$Distance_i$  : Jarak antara data latih ke- $i$  dengan data uji.

$\alpha$  : Smoothing (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.

## Keterangan:

*Voting*( $lbl_x$ ) : Voting dari kelas data latih x.

$n$  : Jumlah data latih dengan kelas data x.

*i* : Nilai indeks.

$W_i$  : Bobot data latih ke-*i*.

## **BAB III**

### **METODOLOGI PENELITIAN**

Pada bab ini, akan dijelaskan mengenai gambaran umum teknis penelitian, data yang akan digunakan dan diolah dalam penelitian, dan keterangan rinci tahap-tahap penelitian.

#### **3.1. Pengumpulan Data**

Dalam penelitian ini, data yang digunakan adalah data sekunder yang diperoleh dari website <https://www.Spotify.com> dengan akses pengambilan data API Spotify menggunakan *Python*. Data yang digunakan dalam penelitian ini berjumlah 50 musik untuk setiap *genre*. *Genre* yang digunakan berjumlah 14 *genre*, 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 di layanan *streaming music* Spotify.

#### **3.2. Variabel dan Definisi Variabel**

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*, *acousticness*, *key*, *mode*, dan *duration*). Fitur *playlist* digunakan sebagai id untuk melabelkan daftar putar lagu untuk *genre* musik. Penjelasan dan definisi dari variabel yang digunakan dalam penelitian ini ditampilkan dalam Tabel 3.1 dan Tabel 3.2.

Tabel 3.1. Variabel Fitur Pengguna

No	Variabel	Definisi Variabel	Ukuran
1.	<i>Playlist</i>	Daftar putar lagu 0 untuk <i>genre</i> musik <i>Blues</i> , 1 untuk <i>genre</i> musik <i>Classical</i> , 2 untuk <i>genre</i> musik <i>Country</i> , 3 untuk <i>genre</i> musik <i>EDM</i> , 4 untuk <i>genre</i> musik <i>Funk</i> , 5 untuk <i>genre</i> musik <i>Funk</i> , 6 untuk	—

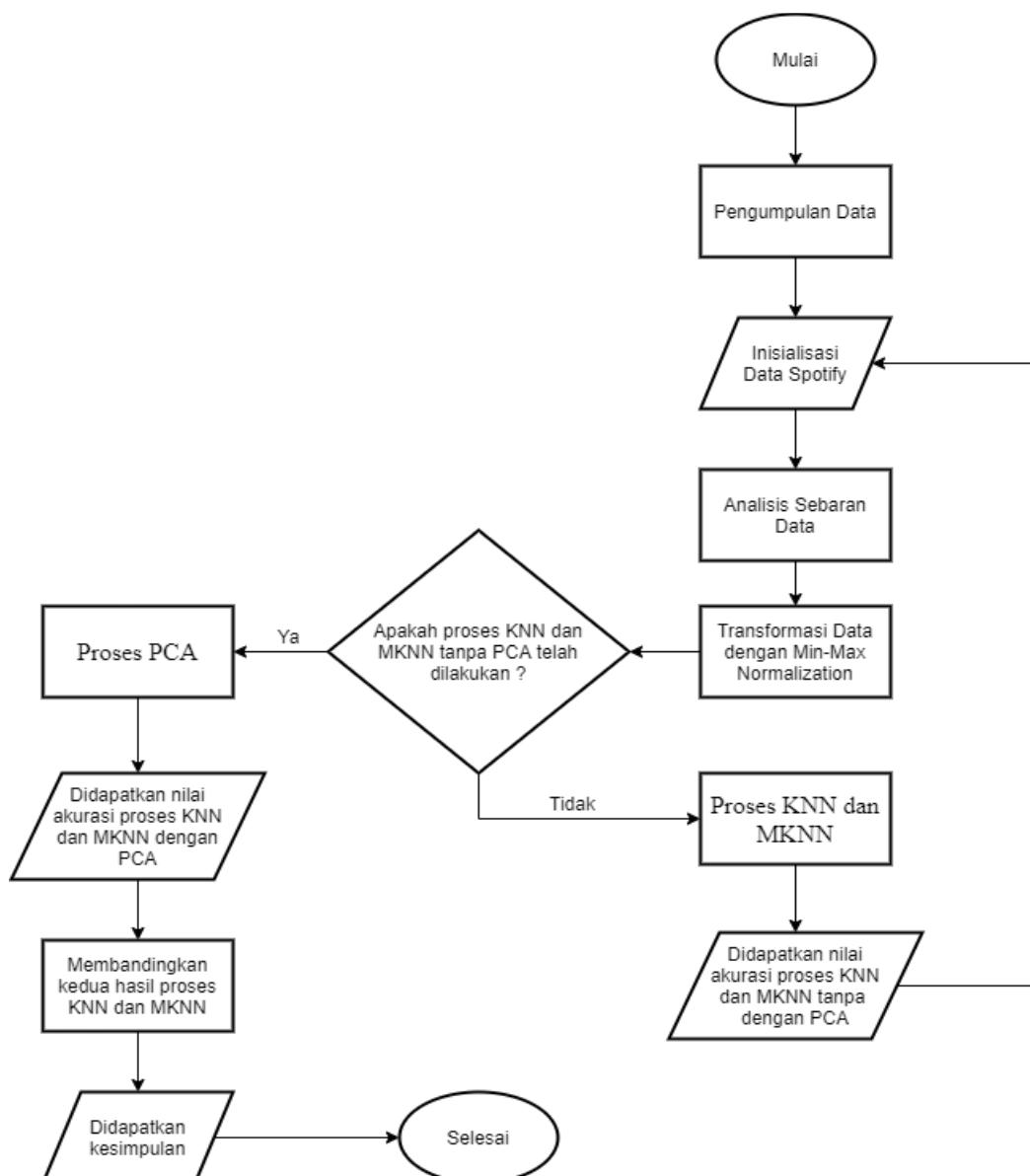
		<i>genre</i> musik <i>Hip Hop</i> , 7 untuk <i>genre</i> musik <i>Jazz</i> , 8 untuk <i>genre</i> musik <i>Metal</i> , 9 untuk <i>genre</i> musik <i>Pop</i> , 10 untuk <i>genre</i> musik <i>R&amp;B</i> , 11 untuk <i>genre</i> musik <i>Reggae</i> , 12 untuk <i>genre</i> musik <i>Rock</i> , dan 13 untuk <i>genre</i> musik <i>Soul</i>	-
--	--	---	---

Tabel 3.2. Variabel Fitur Audio

1.	<i>Danceability</i>	Menjelaskan seberapa kecocokan sebuah lagu atau trek musik untuk menari	0,0 – 1,0
2.	<i>Valence</i>	Mengindikasi jenis emosi (seperti bahagia atau sedih)	0,0 – 1,0
3.	<i>Energy</i>	Ukuran intensitas dan aktivitas suatu tingkat dari kemampuan suatu musik untuk meningkatkan emosi dari pendengarnya	0,0 – 1,0
4.	<i>Tempo</i>	Tempo merupakan cepat atau lambatnya sebuah lagu. Ukuran dari sebuah <i>tempo</i> adalah <i>beats per minute</i> (BPM).	<i>Beats Per Minute</i> (BPM)
5.	<i>Loudness</i>	Kenyaringan (lembut) pada suatu lagu dalam desibel (dB)	-60 – 0 dB
6.	<i>Speechiness</i>	Mendeteksi keberadaan kata-kata yang diucapkan di dalam sebuah trek lagu	> 0,66 = Trek Kata-kata 0,66 – 0,33 = Musik & Kata-kata < 0,33 = Trek NonSpeech
7.	<i>Instrumentalness</i>	Memprediksi apakah suatu lagu mengandung vokal atau tidak, seperti suara “Ooh” dan “Aah”	0,0 – 1,0
8.	<i>Liveness</i>	Mendeteksi keberadaan <i>audiens</i> dalam rekaman	0,0 – 1,0
9.	<i>Acousticness</i>	Suatu ukuran yang menunjukkan apakah trek tersebut akustik atau tidak	0,0 – 1,0

10.	<i>Key</i>	Kunci keseluruhan trek yang diperkirakan. <i>Integer</i> memetakan ke <i>pitch</i> menggunakan <i>standard Pitch Class notation</i>	0 = C, 1 = C#/D b, 2 = D
11.	<i>Mode</i>	Mengindikasikan modalitas (besar atau kecil) dari suatu trek, jenis skala dari mana konten melodinya diturunkan	Mayor diwakili oleh 1 dan minor adalah 0
12.	<i>Duration</i>	Panjang durasi trek musik	Milidetik

### 3.3. Alur Penelitian Sistem Klasifikasi *Genre* Musik



Gambar 3.1. Diagram Alir Sistem Klasifikasi *Genre* Musik

Pada bagian ini akan digambarkan diagram alir sistem klasifikasi *genre* musik dari penelitian yang akan dilakukan penulis seperti pada Gambar 3.1, yaitu dimulai dari inisialisasi data spotify dan analisis sebaran data, transformasi data dengan min-max normalization, proses KNN dan MKNN tanpa PCA, proses KNN dan MKNN dengan PCA, didapatkan nilai akurasi proses KNN dan MKNN, membandingkan kedua hasil proses KNN dan MKNN tanpa PCA dan dengan PCA, didapatkan kesimpulan. Berikut ini definisi penjelasan dari masing-masing tahap alur penelitian pada Gambar 3.1.

Tahap pertama yang dilakukan dalam penelitian ini adalah proses pengumpulan data dengan membuat *playlist* lagu. *Playlist* dibuat untuk masing-masing 14 *genre* pada layanan streaming Spotify, yang kemudian diunduh menggunakan Web API dan disimpan ke dalam *spreadsheet*.

Tahap kedua adalah inisialisasi data spotify dengan memberikan label id pada setiap daftar putar lagu agar bisa membedakan antar *genre* musik.

Tahap ketiga adalah analisis sebaran data untuk melihat *range* data pada variabel fitur audio yang digunakan berdasarkan variabel fitur audio pada Tabel 3.2.

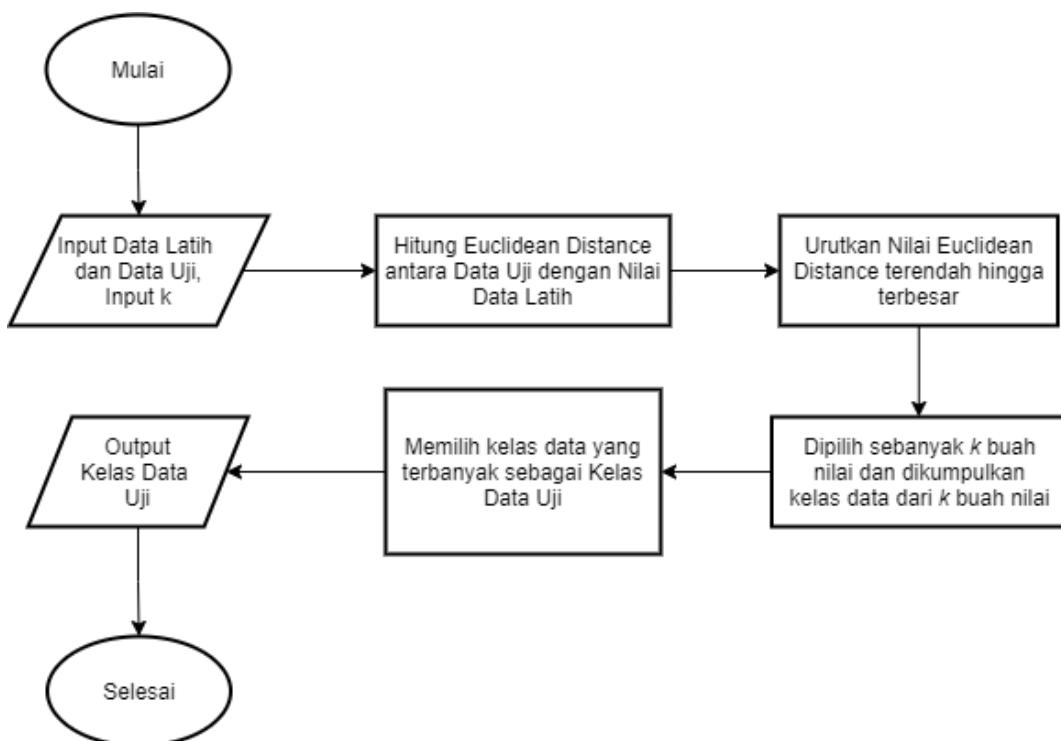
Tahap keempat adalah transformasi data. Transformasi data dilakukan guna mendapatkan *range* data yang sama dari dua variabel data atau lebih yang memiliki *range* berbeda jauh dari variabel yang lainnya. Transformasi yang digunakan adalah *min–max normalization* menggunakan cara pada Persamaan (2.1).

Tahap kelima adalah pengujian. Pengujian akan dilakukan dengan dua pengujian. Pada pengujian pertama yaitu proses tanpa seleksi fitur *Principal Component Analysis* (PCA) dan pengujian kedua yaitu proses menggunakan seleksi fitur *Principal Component Analysis* (PCA). Untuk pengujian pertama dan pengujian kedua, pembagian data latih dan data uji menggunakan metode *k-fold cross-validation*. Pembagian data pada pengujian pertama dan pengujian kedua adalah 90:10%, dengan pembagian jumlah data yaitu 90% data latih sebanyak 630 data lagu dan 10% data uji sebanyak 70 data lagu. Untuk 70 lagu data uji akan diambil secara merata yaitu 5 data lagu pada tiap – tiap *genre* atau tiap kelas data. Setelah itu dilakukan pengujian proses pengklasifikasian menggunakan metode *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan nilai *k* bilangan ganjil agar meningkatkan kecepatan algoritma

dengan menghindari pengklasifikasi genap untuk menghindari kemungkinan dua kelas berbeda memiliki jumlah suara yang sama (Hassanat, 2004). Selama proses pengklasifikasian KNN dan MKNN, dilakukan evaluasi untuk mengukur seberapa tepat klasifikasi tiap kelas dengan *confusion matrix*.

Tahap terakhir adalah kesimpulan. Hasil nilai akurasi pengujian pertama dan pengujian kedua akan dikumpulkan terlebih dahulu. Setelah itu membandingkan metode KNN dan MKNN tanpa seleksi fitur PCA dengan seleksi fitur PCA.

### 3.4. Proses KNN (*K–Nearest Neighbor*)

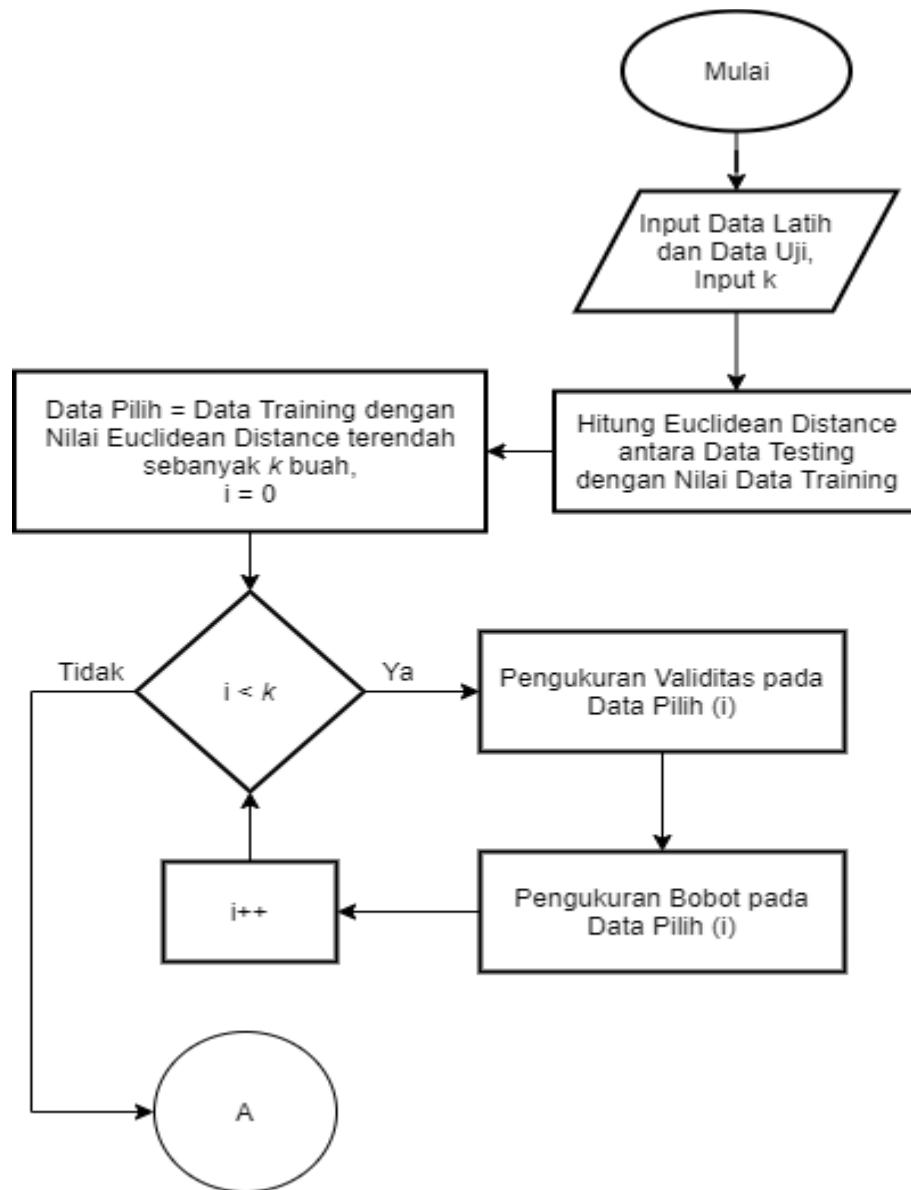


Gambar 3.2. Diagram Alir Proses KNN

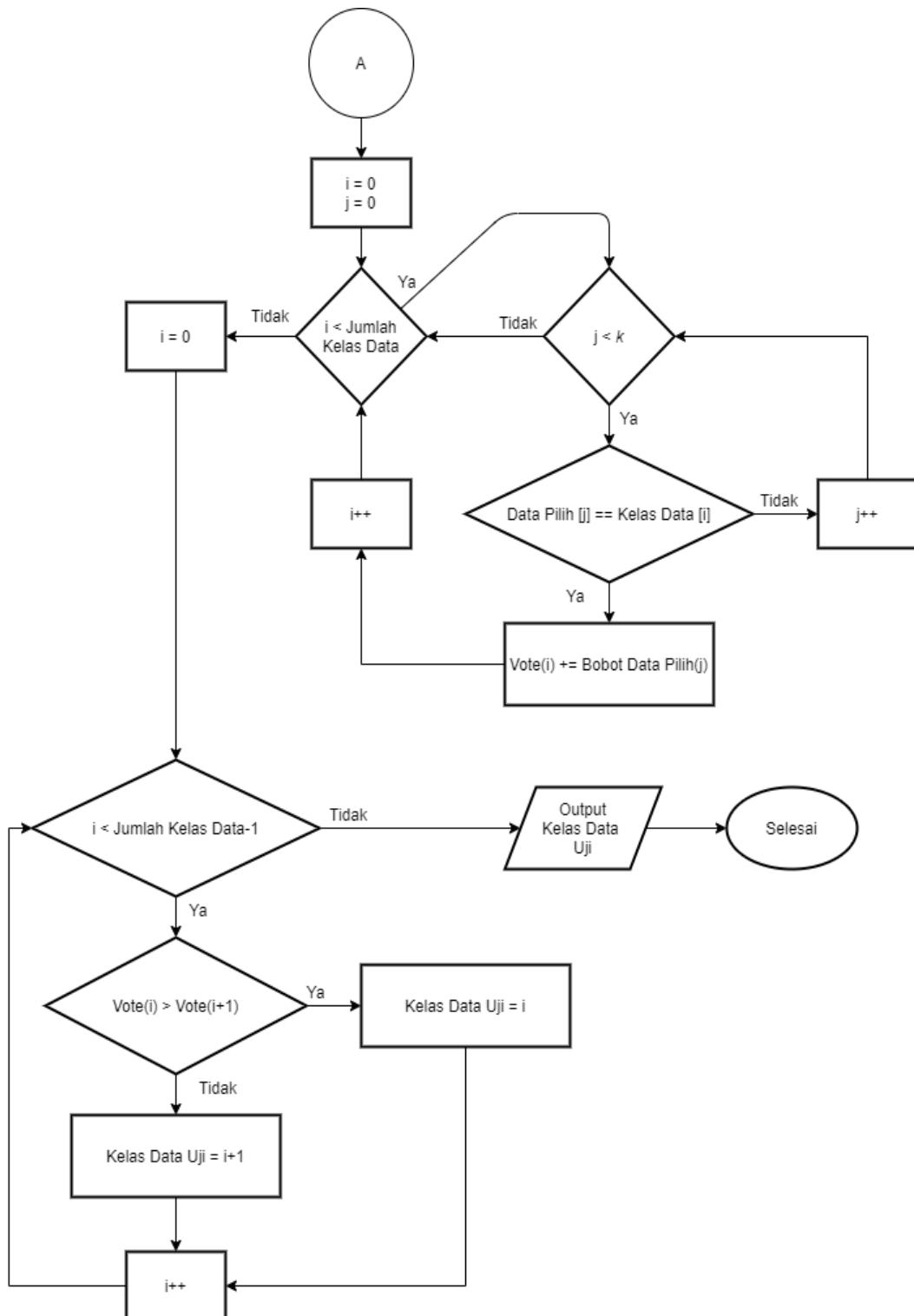
Pada Gambar 3.2 diagram alir proses KNN, dijelaskan tahapan-tahapan proses klasifikasi metode algoritma *K–Nearest Neighbor* (KNN) dalam menyelesaikan keseluruhan penelitian ini. Pada Gambar 3.2 menjelaskan proses data uji yang diinputan. Selanjutnya dihitung jarak berdasarkan nilai *euclidean distance* dengan cara pada Persamaan (2.8) karena nilai *euclidean distance* ini merepresentasikan kedekatan atau kemiripan antar data uji dengan data latih dimana semakin kecil nilai *euclidean distance* maka semakin mirip data tersebut. Kemudian nilai *euclidean distance* akan diurutkan dari terendah hingga terbesar.

Setelah diurutkan berdasarkan nilai *euclidean distance* dari data latih, maka akan dipilih sebanyak  $k$  buah dan dikumpulkan kelas data dari  $k$  buah nilai (tetangga terdekat). Setelah itu memilih kelas data yang terbanyak pertama yang merupakan kelas data dari data uji tersebut, maka data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut.

### 3.5. Proses MKNN (*Modified K–Nearest Neighbor*)



Gambar 3.3. Diagram Alir Proses MKNN



Gambar 3.4. Lanjutan Diagram Alir Proses MKNN

Pada Gambar 3.3 dan 3.4 diagram alir proses MKNN, dijelaskan tahapan-tahapan proses klasifikasi metode algoritma *Modified K-Nearest Neighbor* (MKNN) dalam menyelesaikan keseluruhan penelitian ini. Pada Gambar 3.3

menjelaskan proses data uji yang diinputan. Selanjutnya dihitung jarak berdasarkan nilai *euclidean distance* dengan cara pada Persamaan (2.8). Setelah mendapatkan nilai *euclidean distance* sejumlah nilai  $k$ , selanjutnya dilakukan validasi dengan cara pada Persamaan (2.9) dan Persamaan (2.10) pada data latih lainnya. Dari validasi tersebut didapatkan bobot dengan cara pada Persamaan (2.11) pada masing-masing kelas datanya. Pada Gambar 3.4 dapat dilihat bahwa bobot yang didapatkan akan diurutkan sesuai kelas data. Kemudian pemilihan bobot akan dilakukan dengan cara pada Persamaan (2.12) dengan memilih bobot yang tertinggi yang merupakan kelas data dari data uji tersebut, maka data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut.

### **3.6. Tahap Pengujian dan evaluasi**

Pengujian sistem berfungsi untuk mengetahui akurasi dari sistem itu sendiri dalam melakukan tugas yaitu mengklasifikasi *genre*. Terdapat beberapa tahap pengujian dan evaluasi yang akan dilakukan pada penelitian ini yaitu:

#### **3.6.1. Pengujian Pertama**

Pada pengujian pertama dalam penelitian ini yaitu membandingkan akurasi dari tiap perubahan nilai  $k$  pada *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan metode *k-fold cross-validation* tanpa seleksi fitur *Principal Component Analysis* (PCA). Perhitungan akurasi dilakukan dengan  $k$  bernilai bilangan ganjil kisaran 3 sampai dengan 29. Pembagian data latih dan data uji yang dilakukan dengan menggunakan metode *k-fold cross-validation*. Pembagian data pada pengujian pertama dan pengujian kedua adalah 90:10%, dengan pembagian jumlah data yaitu 90% data latih sebanyak 630 data lagu dan 10% data uji sebanyak 70 data lagu. Untuk 70 lagu data uji akan diambil secara merata yaitu 5 data lagu pada tiap – tiap *genre* atau tiap kelas data.

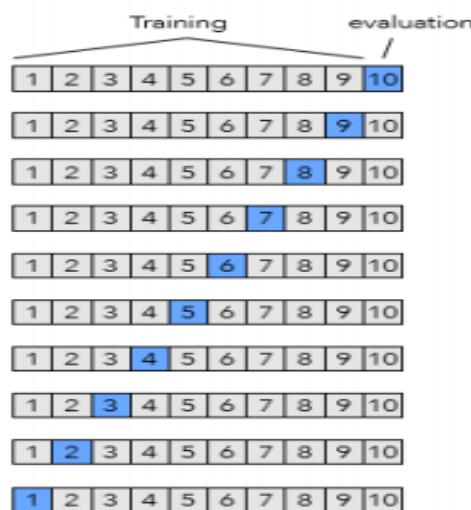
#### **3.6.2. Pengujian Kedua**

Pada pengujian kedua dalam penelitian ini yaitu membandingkan akurasi dari tiap perubahan nilai  $k$  pada *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan seleksi fitur *Principal Component Analysis*

(PCA). Perhitungan akurasi dilakukan dengan  $k$  bernilai bilangan ganjil kisaran 3 sampai dengan 29. Pembagian data latih dan data uji yang dilakukan dengan menggunakan metode  $k$ -*fold cross-validation*. Pembagian data pada pengujian pertama dan pengujian kedua adalah 90:10%, dengan pembagian jumlah data yaitu 90% data latih sebanyak 630 data lagu dan 10% data uji sebanyak 70 data lagu. Untuk 70 lagu data uji akan diambil secara merata yaitu 5 data lagu pada tiap – tiap *genre* atau tiap kelas data.

### 3.6.3. Evaluasi

Pada tahap pengujian pertama dan pengujian kedua ini akan sama – sama digunakan pengujian menggunakan metode  $k$ -*fold cross-validation*. Dalam  $k$ -*fold cross-validation*, data secara acak akan dipartisi menjadi  $k$  atau “lipatan” yang masing–masing berukuran sama. Sehingga, data data uji dan data uji akan dilakukan sebanyak  $k$  kali. Dalam iterasi pertama,  $d_1$  dipilih sebagai data uji, dan partisi yang tersisa yaitu  $d_2, \dots, d_k$  secara kolektif digunakan sebagai data latih. Sedangkan dalam iterasi kedua, partisi  $d_2$  dipilih sebagai data uji, dan partisi  $d_1, d_3, \dots, d_k$  digunakan sebagai data latih, dan begitu seterusnya sampai iterasi sebanyak  $k$ -*fold*, kemudian setiap iterasi dihitung berdasarkan nilai akurasi dengan menggunakan *confusion matrix* untuk mengukur seberapa tepat klasifikasi terhadap kelas. Rumus yang digunakan yaitu pada Persamaan (2.7) untuk perhitungan akurasi.



Gambar 3.5. 10–*Fold Cross–Validation*

Untuk kasus klasifikasi menggunakan *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN), tiap iterasi *k–fold* data uji akan dibandingkan dengan data latih untuk mendapatkan nilai akurasi, sehingga nilai rata–rata akurasi dapat diukur berdasarkan jumlah keseluruhan total akurasi tiap *k–fold* dibagi dengan jumlah total *k–fold* yang digunakan. Rumus yang digunakan sebagai berikut (lihat pada Persamaan 2.14).

$$\text{Akurasi} = \frac{\text{Total Akurasi tiap } k\text{-fold}}{\text{Total } k\text{-fold}} \times 100 \quad \dots\dots\dots (2.14)$$

Pada Persamaan (2.14), untuk nilai total *k–fold* yang digunakan pada metode ini adalah  $k = 10$  sesuai dengan (Wibowo, 2017) yaitu *10–Fold Cross–Validation* yang umum digunakan pada Gambar 3.5.

### **3.7. Implementasi Sistem**

Sistem diimplementasikan menggunakan bahasa pemrograman web *PHP* dan database XAMPP versi 3.2.4 dalam menyimpan dataset. Penelitian ini menggunakan editor kode IDE Visual Studio Code yang dijalankan dengan menggunakan Sistem Operasi Windows 10 64-bit.

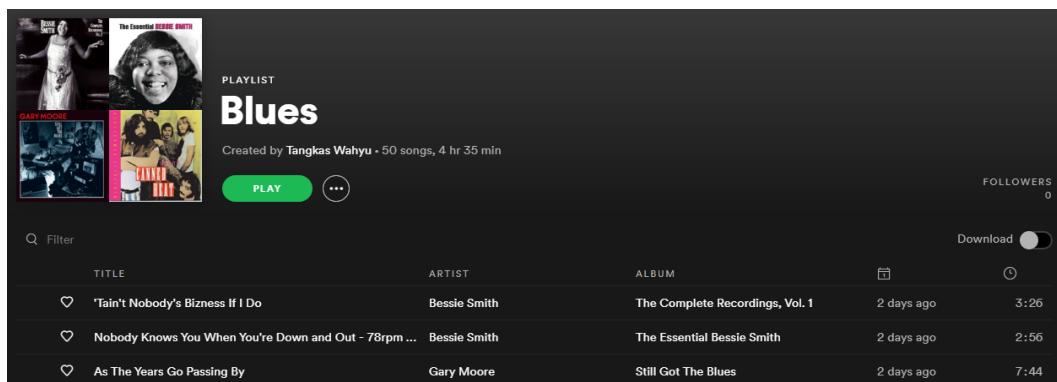
## BAB IV

# HASIL DAN PEMBAHASAN

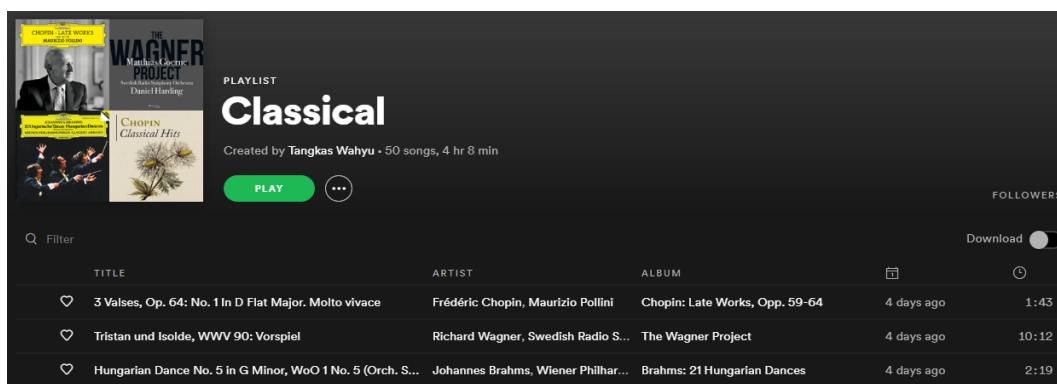
Pada bab ini, akan dijelaskan mengenai hasil pengaruh atribut lagu terhadap kinerja KNN dan MKNN, transformasi data, implementasi KNN dan MKNN, implementasi sistem, pengujian pertama klasifikasi *genre* tanpa seleksi fitur, pengujian kedua dengan seleksi fitur dan analisis hasil pengujian.

### 4.1. Pengumpulan Data

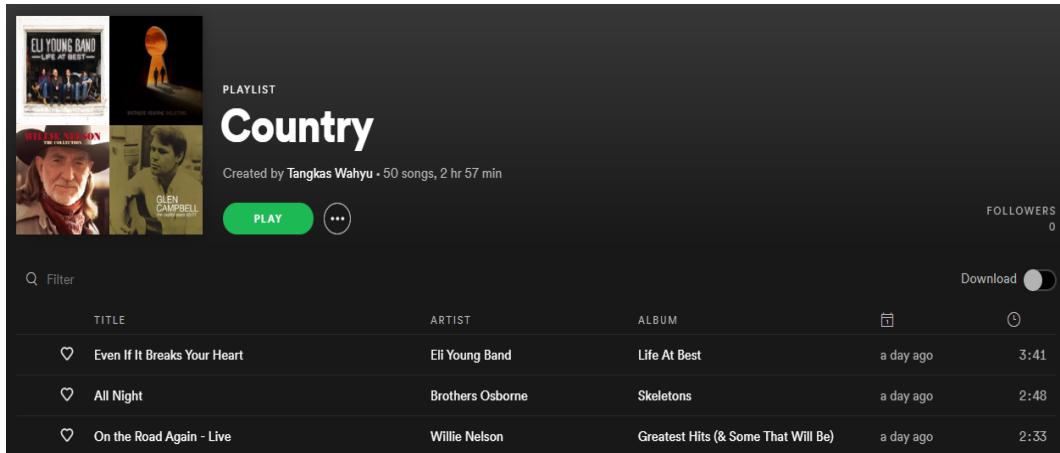
Spotify digunakan sebagai platform dalam membuat *playlist genre* berisi lagu. Pada penelitian ini, *playlist* baru dibuat agar dapat dibedakan dengan *playlist* yang telah ada. Sebuah lagu kemudian diambil pada *playlist* yang sudah disediakan oleh Spotify. Berikut adalah *playlist* 14 *genre* musik dan data lagu yang di gunakan dari website <https://www.Spotify.com>:



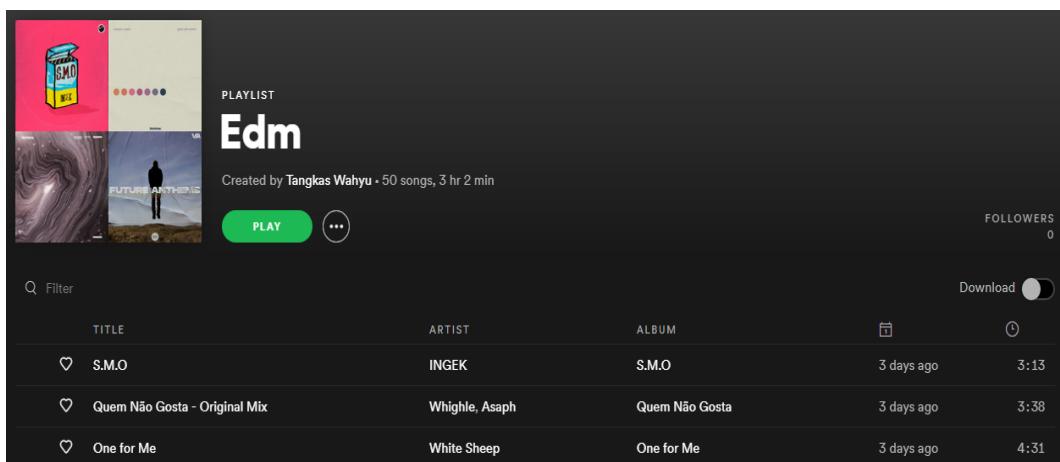
Gambar 4.1. *Playlist Genre Blues*



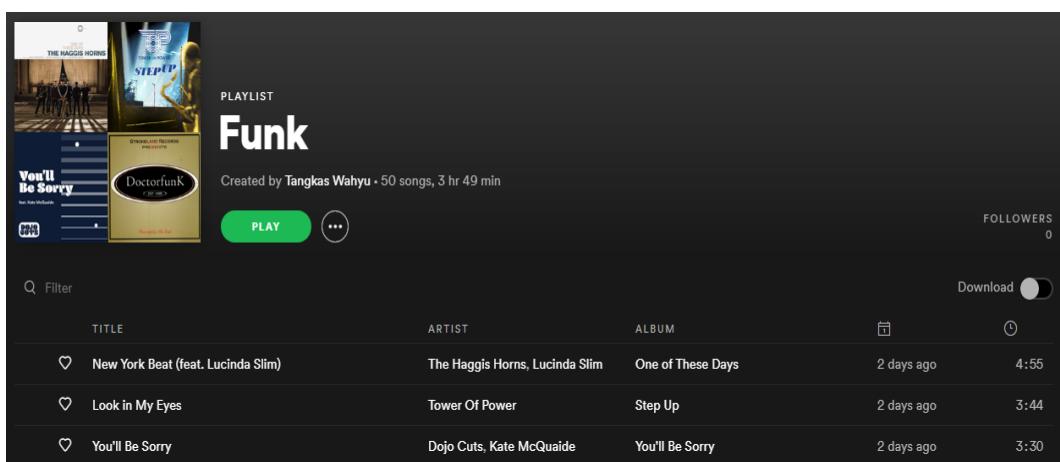
Gambar 4.2. *Playlist Genre Classical*



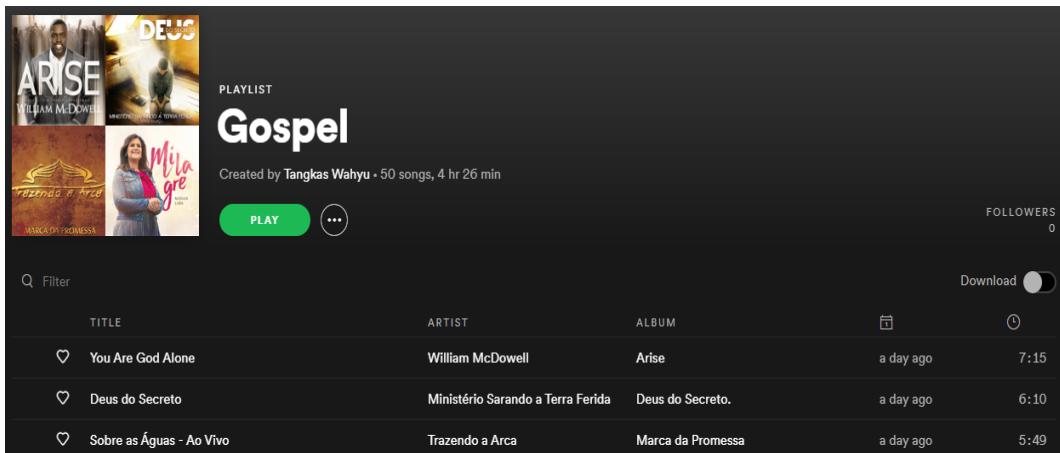
Gambar 4.3. Playlist Genre Country



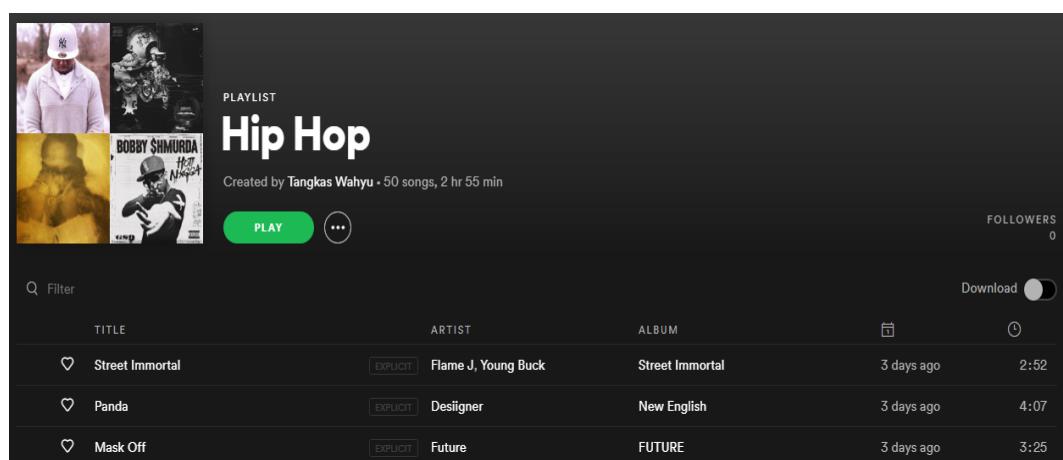
Gambar 4.4. Playlist Genre EDM



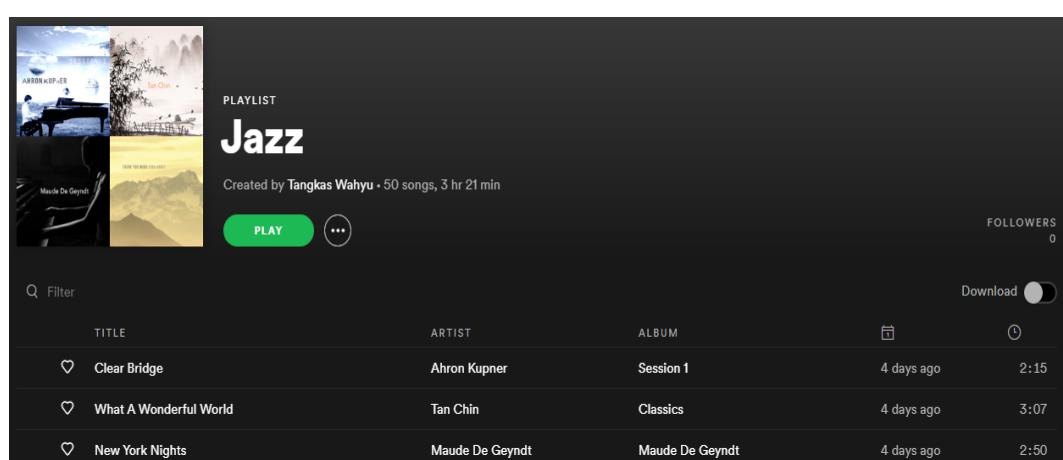
Gambar 4.5. Playlist Genre Funk



Gambar 4.6. Playlist Genre Gospel



Gambar 4.7. Playlist Genre Hip Hop



Gambar 4.8. Playlist Genre Jazz

**PLAYLIST**  
**Metal**  
Created by Tangkas Wahyu • 50 songs, 3 hr 20 min

**PLAY** • **...**

**FOLLOWERS** 0

**Q Filter** **Download**

TITLE	ARTIST	ALBUM	DATE	TIME
Pixelate	Volumes	Pixelate	3 days ago	3:10
An Epilogue to the Arrogant	Drown in Sulphur	An Epilogue to the Arrogant	3 days ago	4:06
Human Swarm	Humanity's Last Breath	Humanity's Last Breath	3 days ago	4:25

Gambar 4.9. *Playlist Genre Metal*

**PLAYLIST**  
**Pop**  
Created by Tangkas Wahyu • 50 songs, 3 hr 20 min

**PLAY** • **...**

**FOLLOWERS** 0

**Q Filter** **Download**

TITLE	ARTIST	ALBUM	DATE	TIME
A Thousand Miles	Vanessa Carlton	Be Not Nobody	8 hours ago	3:57
If Tomorrow Never Comes	Ronan Keating	Destination	8 hours ago	3:34
I'm Yours	Jason Mraz	We Sing. We Dance. We Steal Thi...	8 hours ago	4:02

Gambar 4.10. *Playlist Genre Pop*

**PLAYLIST**  
**R&B**  
Created by Tangkas Wahyu • 50 songs, 3 hr 13 min

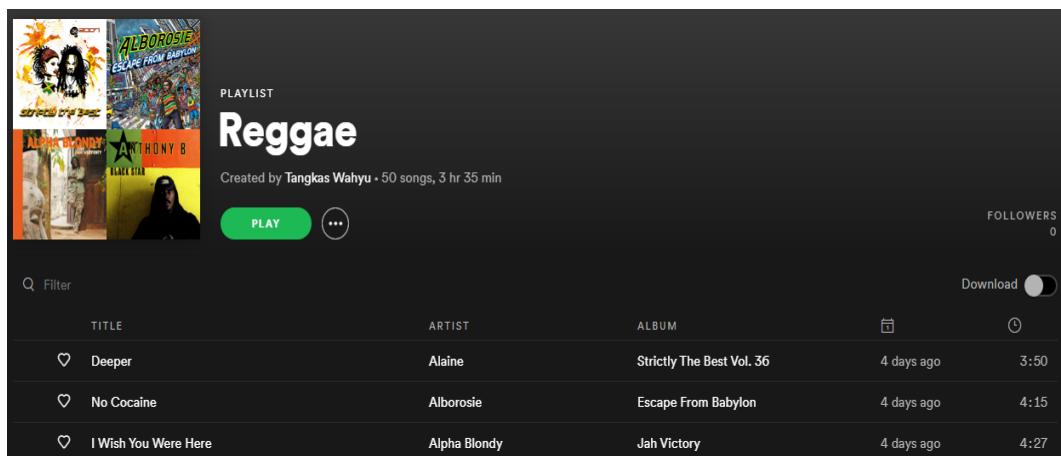
**PLAY** • **...**

**FOLLOWERS** 0

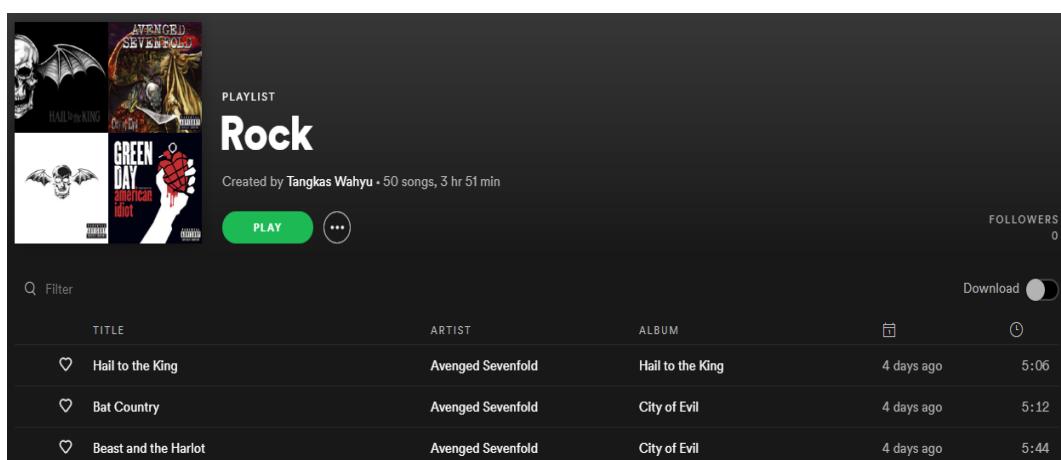
**Q Filter** **Download**

TITLE	ARTIST	ALBUM	DATE	TIME
You Can't Judge A Book By It's Cover	Bo Diddley	The Chess Box	3 days ago	3:11
I'll Take You There	The Staple Singers	The Very Best Of The Staple Singers	3 days ago	3:17
B-A-B-Y	Carla Thomas	Carla	3 days ago	2:57

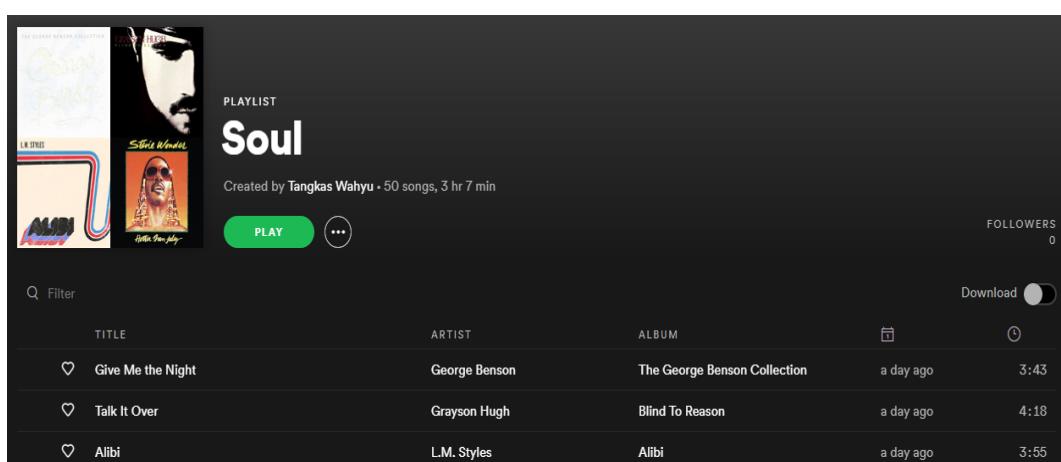
Gambar 4.11. *Playlist Genre R&B*



Gambar 4.12. Playlist Genre Reggae



Gambar 4.13. Playlist Genre Rock



Gambar 4.14. Playlist Genre Soul

Pengunduhan data diambil dari Web *Application Programming Interface* (API) Spotify melalui Bahasa Pemrograman Python versi 3.8.2. Syntax yang digunakan untuk mempermudah mengunduh data adalah sebagai berikut:

```
# Import libraries
import pandas as pd
import spotipy
import os
from spotipy.oauth2 import SpotifyClientCredentials

# Set Web API
os.environ['SPOTIPY_CLIENT_ID'] = '9a8cb565889548e69d001505e9940d4
b'
os.environ['SPOTIPY_CLIENT_SECRET'] = 'c251cd9f1eaf427081525bf4bd7
74bea'

sp = spotipy.Spotify(client_credentials_manager=SpotifyClientCred
entials()) # Set sp = Spotipy
playlist = sp.playlist('masukan url
playlist', fields=None, market=None, additional_types=('track', ))
# URL Spotify Playlist Genre
tracks = playlist['tracks']['items'] # Set Track = Playlist

track_list = []
for i in range(len(tracks)):
    track_list.append(tracks[i]['track']['uri'])

features = sp.audio_features(tracks=track_list) # Set Features = s
p.audio_features(Panjang track = track_list)
pd.DataFrame(features).to_csv('Lokasi Penyimpanan dan nama
file.csv') # Save to spreadsheet format .csv
```

## 4.2. Inisialisasi Data Spotify

14 *playlist* yang telah dibuat akan diberikan label id pada setiap daftar putar lagu agar bisa membedakan antar *genre* musik. 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. Masing-masing *playlist* berisi 50 lagu. Lagu-lagu tersebut, yang telah disediakan oleh Spotify pada Lampiran A.1 sampai dengan Lampiran A.14, sehingga total lagu yang digunakan pada penelitian ini adalah 700 lagu. Berikut data lagu yang telah diunduh melalui web API Spotify ditunjukkan pada Tabel 4.1.

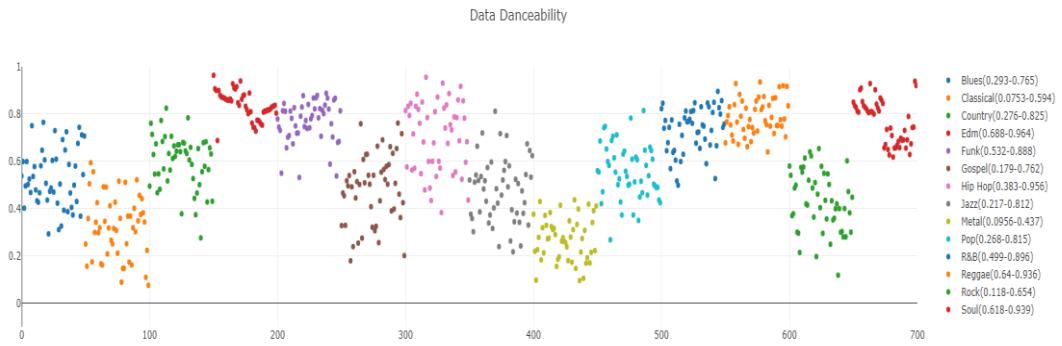
Tabel 4.1. Data *Playlist* Musik

No	Playlist	Danceability	Energy	Key	Loudness	Mode	Speechiness	Acousticness
1	0	0.538	0.0466	3	-16.466	0	0.0349	0.996
2	0	0.599	0.0424	4	-12.099	1	0.0572	0.996
3	0	0.402	0.0982	5	-19.814	0	0.0401	0.356
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
698	13	0.745	0.272	2	-16.539	0	0.0551	0.561
699	13	0.939	0.487	10	-7.641	0	0.153	0.0172
700	13	0.921	0.529	2	-11.974	1	0.0851	0.0845

No	Playlist	Instrumentalness	Liveness	Valence	Tempo	Duration
1	0	0.000606	0.212	0.141	80.659	206227
2	0	0.00176	0.185	0.186	90.001	176133
3	0	0.194	0.358	0.191	141.54	464040
-	-	-	-	-	-	-
-	-	-	-	-	-	-
698	13	0.0159	0.0529	0.504	101.947	199400
699	13	0.0124	0.05	0.646	117.637	359280
700	13	0.000779	0.0478	0.724	102.941	334360

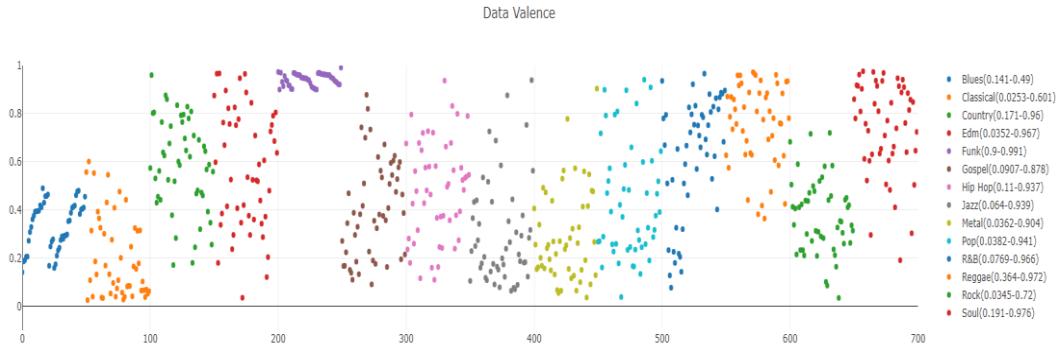
### 4.3. Analisis Sebaran Data

Berdasarkan pada Tabel 4.1 akan dibuatkan plot sebaran data untuk melihat *range* data pada variabel fitur audio yang digunakan. Berikut adalah plot sebaran data dari variabel fitur audio yang digunakan:



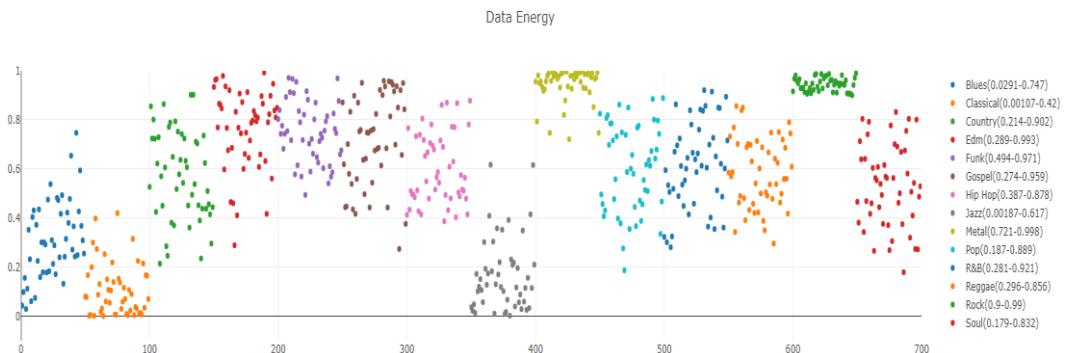
Gambar 4.15. Plot sebaran data *Danceability*

Pada Gambar 4.15, terlihat bahwa sebaran data dari variabel fitur *danceability* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



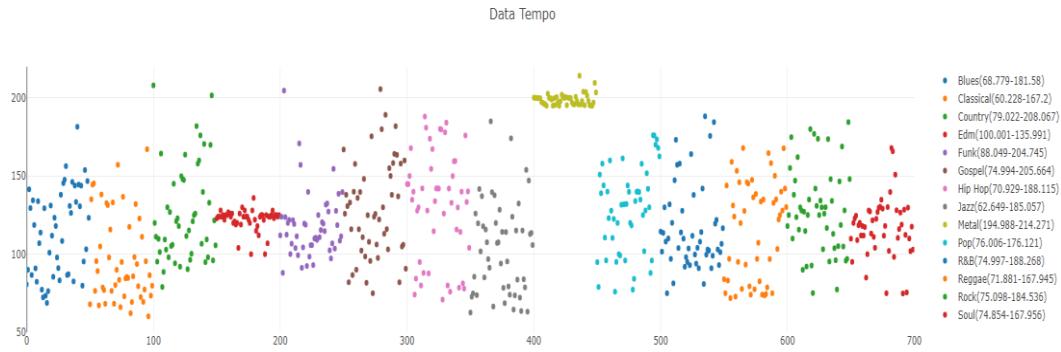
Gambar 4.16. Plot sebaran data *Valence*

Pada Gambar 4.16, terlihat bahwa sebaran data dari variabel fitur *valence* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



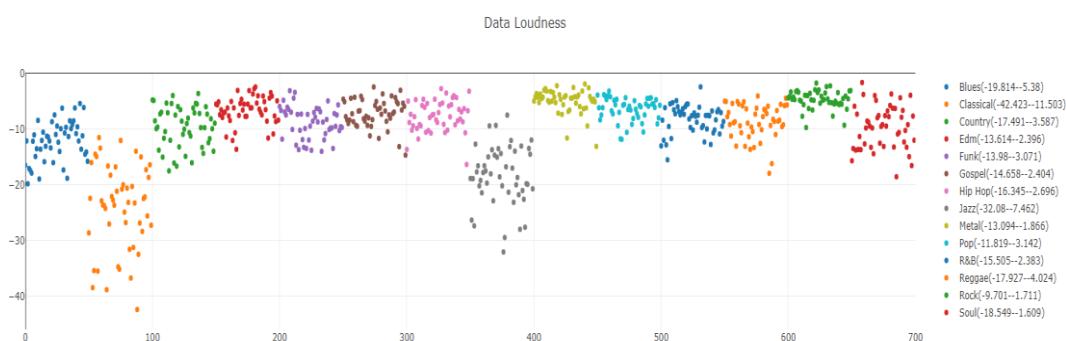
Gambar 4.17. Plot sebaran data *Energy*

Pada Gambar 4.17, terlihat bahwa sebaran data dari variabel fitur *Energy* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



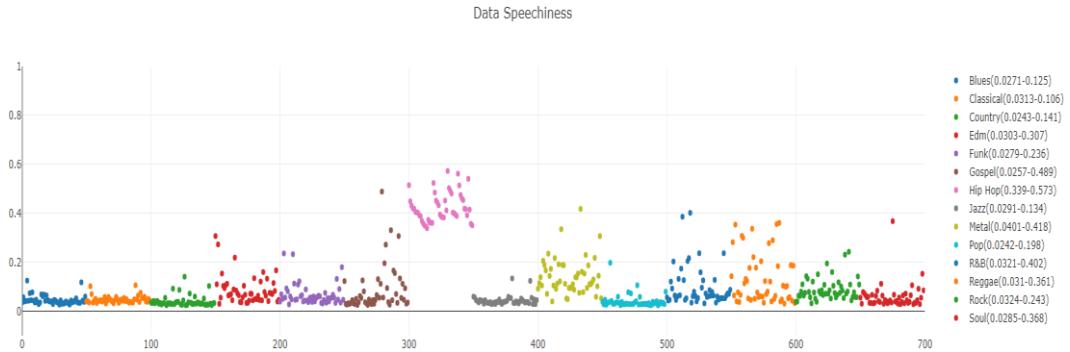
Gambar 4.18. Plot sebaran data *Tempo*

Pada Gambar 4.18, terlihat bahwa sebaran data dari variabel fitur *Tempo* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 62.649 *beats per minute* (BPM) hingga 214.271 *beats per minute* (BPM), sehingga variabel fitur *Tempo* memiliki rentang *range* data berbeda dari variabel fitur lainnya.



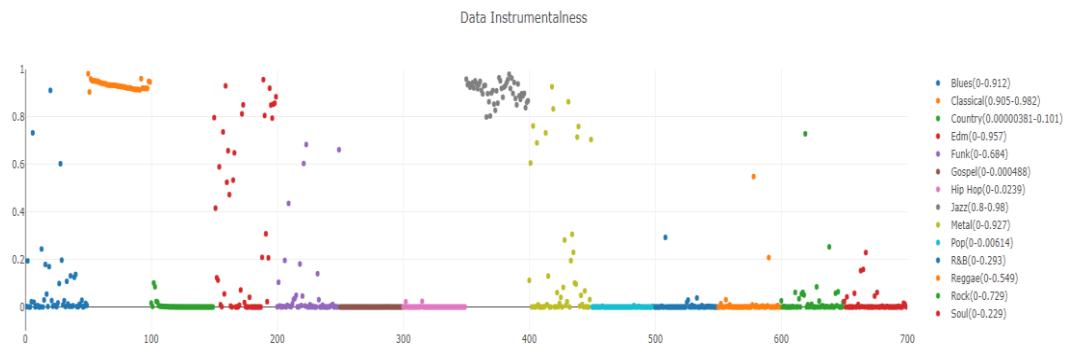
Gambar 4.19. Plot sebaran data *Loudness*

Pada Gambar 4.19, terlihat bahwa sebaran data dari variabel fitur *Loudness* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari -32.08 desibel (dB) hingga -1.069 desibel (dB), sehingga variabel fitur *Loudness* memiliki rentang *range* data berbeda dari variabel fitur lainnya.



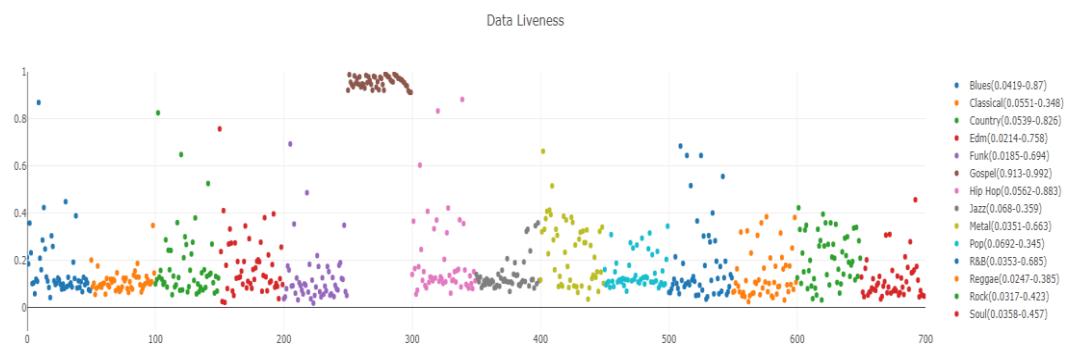
Gambar 4.20. Plot sebaran data *Speechiness*

Pada Gambar 4.20, terlihat bahwa sebaran data dari variabel fitur *Speechiness* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



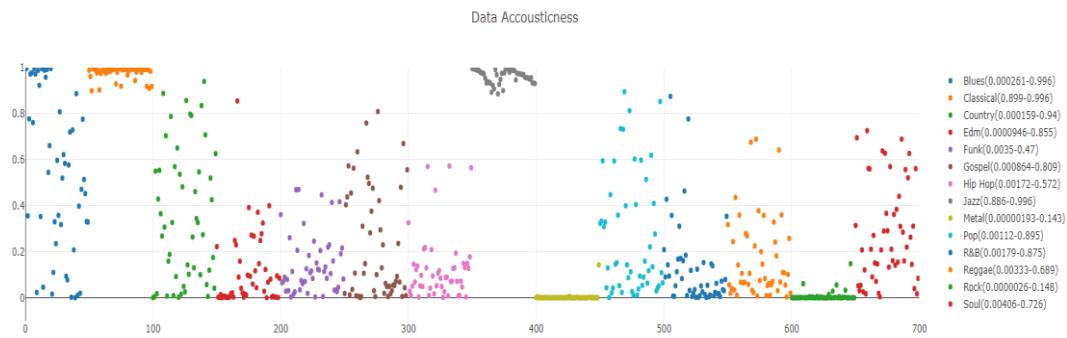
Gambar 4.21. Plot sebaran data *Instrumentalness*

Pada Gambar 4.21, terlihat bahwa sebaran data dari variabel fitur *Instrumentalness* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



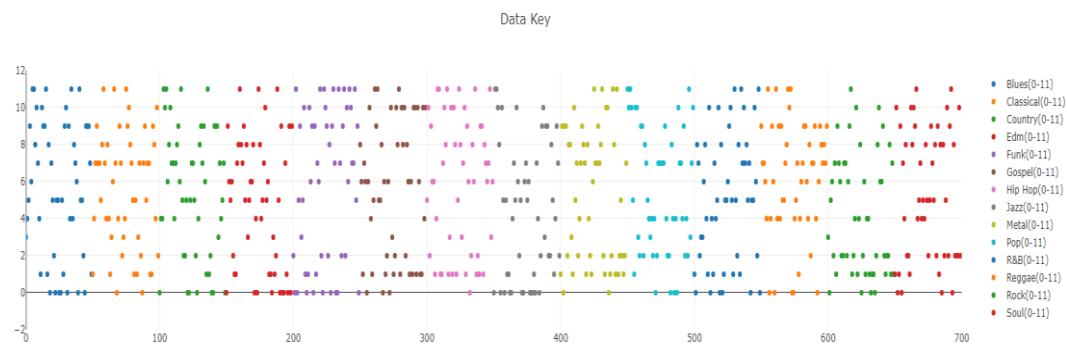
Gambar 4.22. Plot sebaran data *Liveness*

Pada Gambar 4.22 sebelumnya, terlihat bahwa sebaran data dari variabel fitur *Liveness* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



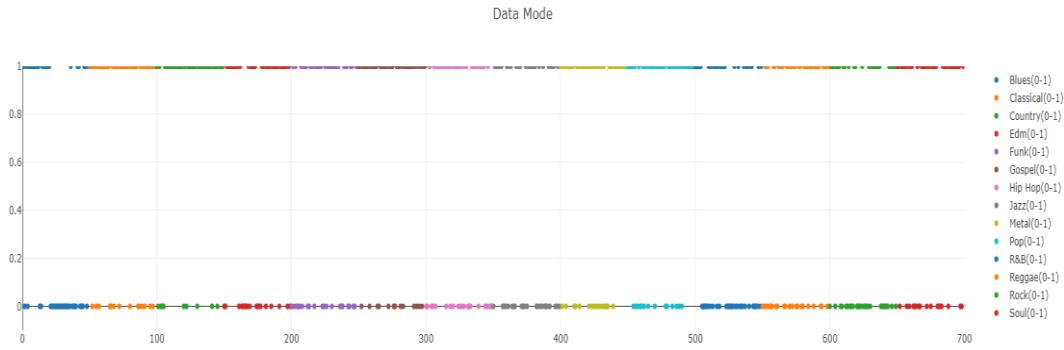
Gambar 4.23. Plot sebaran data *Accousticness*

Pada Gambar 4.23, terlihat bahwa sebaran data dari variabel fitur *Accousticness* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



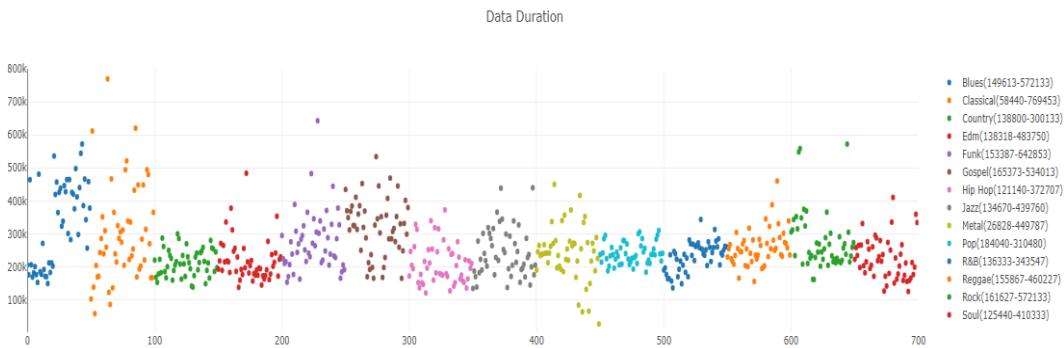
Gambar 4.24. Plot sebaran data *Key*

Pada Gambar 4.24, terlihat bahwa sebaran data dari variabel fitur *Key* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 1 hingga 12 sesuai dengan jumlah trek kunci *standard Pitch Class notation*, sehingga variabel fitur *Key* memiliki rentang *range* data berbeda dari variabel fitur lainnya.



Gambar 4.25. Plot sebaran data *Mode*

Pada Gambar 4.25, terlihat bahwa sebaran data dari variabel fitur *Accousticness* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 0.0 hingga 1.0.



Gambar 4.26. Plot sebaran data *Duration*

Pada Gambar 4.26, terlihat bahwa sebaran data dari variabel fitur *Duration* pada genre musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki rentang *range* data dari 26.828 hingga 642.853 sesuai dengan satuan milidetik, sehingga variabel fitur *Duration* memiliki rentang *range* data berbeda dari variabel fitur lainnya.

Dari seluruh sebaran data tersebut, variabel fitur audio *danceability*, *valence*, *energy*, *speechiness*, *instrumentalness*, *liveness*, *accousticness*, dan *mode* memiliki rentang *range* yang sama dari 0.0 hingga 1.0, sedangkan variabel fitur audio *tempo*, *loudness*, *key* dan *duration* memiliki rentang *range* berbeda. Maka dari itu variabel fitur audio yang akan ditransformasi data adalah variabel fitur *tempo* pada Gambar 4.18, *loudness* pada Gambar 4.19, *key* pada Gambar 4.24, dan *duration* pada Gambar 4.26.

#### 4.4. Transformasi Data

Transformasi data dilakukan karena terdapat empat variabel fitur pada data yang memiliki *range* yang jauh dari skala 0.0 – 1.0 dari variabel fitur lainnya. Variabel fitur yang akan di normalisaikan adalah variabel fitur *tempo* pada Gambar 4.18, *loudness* pada Gambar 4.19, *key* pada Gambar 4.24, dan *duration* pada Gambar 4.26. Variabel fitur tersebut perlu dilakukan transformasi agar range datanya tidak jauh berbeda dengan variabel fitur yang lain, sehingga diharapkan dapat memperoleh hasil penelitian yang lebih baik. Transformasi yang digunakan adalah *min–max normalization*, hal ini dikarenakan agar data hasil transformasi linier dengan data asli dan range yang digunakan adalah 0.0 – 1.0. Rumus min–max normalization dapat dilihat pada Persamaan (2.1). Dapat dilihat pada Tabel 4.2 merupakan data sebelum dilakukan transformasi.

Tabel 4.2. Data Variabel Fitur sebelum ditransformasi

No	Playlist	Key	Loudness	Tempo	Duration
1	0	3	-16.466	80.659	206227
2	0	4	-12.099	90.001	176133
3	0	5	-19.814	141.54	464040
-	-	-	-	-	-
-	-	-	-	-	-
698	13	2	-16.539	101.947	199400
699	13	10	-7.641	117.637	359280
700	13	2	-11.974	102.941	334360

Berikut merupakan contoh penyelesaian normalisasi pada data pertama untuk variabel fitur *loudness*,

Diketahui:

$$V = -16.446$$

$$Min = -42.423$$

$$Max = -1.609$$

$$New\_min = 0$$

$$New\_max = 1$$

$$V' = \frac{-16.446 - (-42.423)}{-1.609 - (-42.423)} (1 - 0) + 0$$

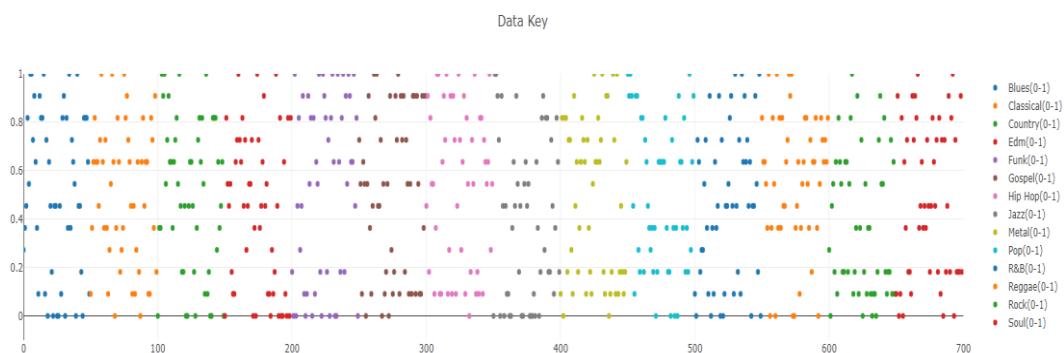
$$V' = 0.635983$$

Diperoleh hasil transformasi untuk data pertama pada variabel fitur *loudness* adalah 0.635983. Cara manual tersebut berlaku juga untuk mentransformasi pada data selanjutnya pada data variabel fitur *key*, *tempo*, *duration*. Hasil dari proses transformasi data dapat dilihat pada Tabel 4.3.

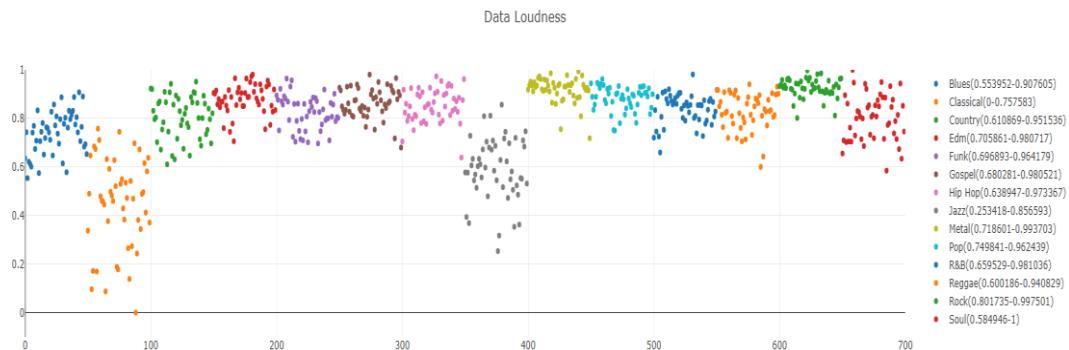
Tabel 4.3. Data Variabel Fitur setelah ditransformasi

No	Playlist	Key	Loudness	Tempo	Duration
<b>1</b>	0	0.27273	0.635983	0.13263	0.241574
<b>2</b>	0	0.36364	0.74298	0.19328	0.20105
<b>3</b>	0	0.45455	0.553952	0.52785	0.588739
-	-	-	-	-	-
-	-	-	-	-	-
<b>698</b>	13	0.18182	0.634194	0.27083	0.232381
<b>699</b>	13	0.90909	0.852208	0.37268	0.447671
<b>700</b>	13	0.18182	0.746043	0.27728	0.414115

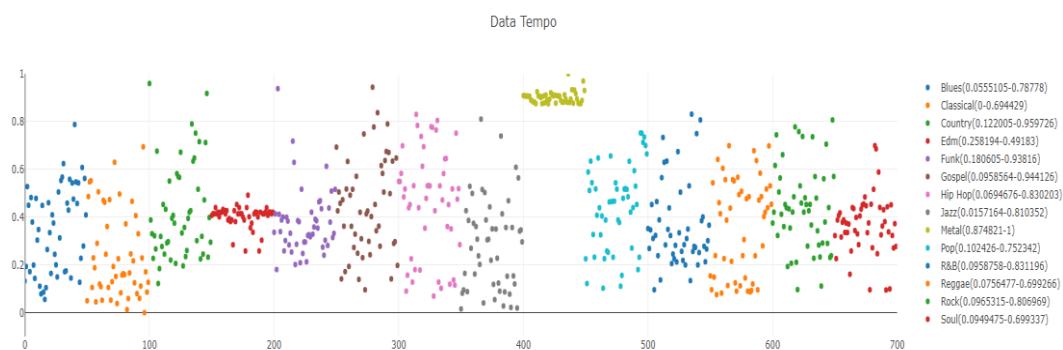
Jika dilihat melalui sebaran data maka *range* data variabel fitur *key*, *loudness*, *tempo*, dan *duration* seperti gambar dibawah ini:



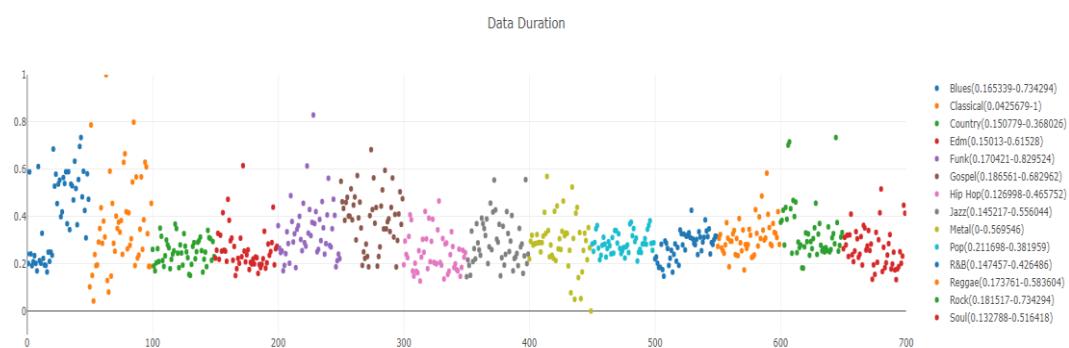
Gambar 4.27. Sebaran data *Key* setelah ditransformasi



Gambar 4.28. Sebaran data *Loudness* setelah ditransformasi



Gambar 4.29. Sebaran data *Tempo* setelah ditransformasi



Gambar 4.30. Sebaran data *Duration* setelah ditransformasi

Berikut ini adalah tabel data *playlist* musik yang telah ditransformasi disajikan pada Tabel 4.4.

Tabel 4.4. Data Lagu 14 *Genre* setelah ditransformasi

-	-	-	-	-	-	-	-	-	-
<b>698</b>	13	0.745	0.272	0.181818	0.634194	0	0.0551	0.561	
<b>699</b>	13	0.939	0.487	0.909091	0.852208	0	0.153	0.0172	
<b>700</b>	13	0.921	0.529	0.181818	0.746043	1	0.0851	0.0845	

No	Playlist	Instrumentalness	Live ness	Vale nce	Tempo	Duration
<b>1</b>	0	0.000606	0.212	0.141	0.132632	0.241574
<b>2</b>	0	0.00176	0.185	0.186	0.193277	0.20105
<b>3</b>	0	0.194	0.358	0.191	0.527853	0.588739
-	-	-	-	-	-	-
-	-	-	-	-	-	-
<b>698</b>	13	0.0159	0.0529	0.504	0.270827	0.232381
<b>699</b>	13	0.0124	0.05	0.646	0.372682	0.447671
<b>700</b>	13	0.000779	0.0478	0.724	0.27728	0.414115

#### 4.5. Implementasi KNN dan MKNN

Metode yang diajukan pada penelitian ini adalah MKNN. MKNN merupakan metode hasil modifikasi dari KNN. Sehingga untuk mengetahui keakuratan dari metode MKNN, maka perlu komparasi antara akurasi dari KNN dan MKNN.

##### 4.3.1. KNN

Pada pengujian pertama dan tahap kedua, bahasa pemrograman yang digunakan pada implementasi metode KNN adalah PHP. Dipilih data uji sebanyak 5 data lagu pada tiap-tiap *genre* atau tiap kelas data, menyeleksi data *genre* dari kelas data latih yang memiliki ID sama dengan data uji ditunjukkan pada baris ke 3 sampai dengan 164 pada Lampiran bagian B.1 untuk pengujian pertama, sedangkan pada pengujian kedua pada Lampiran bagian C.2 baris ke 4 sampai dengan 165. Kemudian Persamaan (2.9) digunakan pada data latih untuk mendapatkan perhitungan jarak antara data latih dengan data uji. Proses tersebut ada pada baris ke 165 sampai dengan 196 pada Lampiran bagian B.1 untuk pengujian pertama, sedangkan pada pengujian kedua pada Lampiran bagian C.2 baris ke 166 sampai dengan 190.

Dari sejumlah  $k$  data latih terpendek, kelas data yang terbanyak muncul menjadi hasil kelas data dari data uji, maka data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut. Proses tersebut terletak pada baris ke 198 sampai

dengan 213 pada Lampiran bagian B.1 pengujian pertama, sedangkan pada pengujian kedua proses tersebut berada pada baris ke 192 sampai dengan 207 pada Lampiran bagian C.2.

#### 4.3.2. MKNN

Pada pengujian pertama dan tahap kedua, bahasa pemrograman yang digunakan pada implementasi metode MKNN adalah PHP. Dipilih data uji sebanyak 5 data lagu pada tiap-tiap *genre* atau tiap kelas data, menyeleksi data *genre* dari kelas data latih yang memiliki ID sama dengan data uji ditunjukkan pada baris ke 3 sampai dengan 164, pada Lampiran bagian B.2 untuk pengujian pertama, sedangkan pengujian kedua pada Lampiran bagian C.3 baris ke 4 sampai dengan 165. selanjutnya Persamaan (2.9) digunakan pada data latih untuk mendapatkan jarak antara data latih dengan data uji yang ditujukan pada baris ke 165 sampai dengan 196, pada Lampiran bagian B.2 untuk pengujian pertama, sedangkan pengujian kedua pada Lampiran bagian C.3 baris ke 166 sampai dengan 190.

Dari setiap data latih yang dipilih akan dilakukan validasi sesuai Persamaan (2.10) dan Persamaan(2.11). Proses tersebut terdapat pada baris 198 sampai dengan 243 pada Lampiran bagian B.2 untuk pengujian pertama. Selanjutnya setiap kelas data hasil validasi dilakukan pembobotan sesuai Persamaan (2.12). Pembobotan terdapat pada baris ke 244 sampai dengan 251 pada Lampiran bagian B.2 untuk pengujian pertama. Selanjutnya kelas data dengan bobot tertinggi menjadi hasil kelas data dari data uji tersebut, maka Data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut sesuai Persamaan (2.13), yang ditujukan pada baris ke 252 sampai dengan 264 pada Lampiran bagian B.2 untuk pengujian pertama.

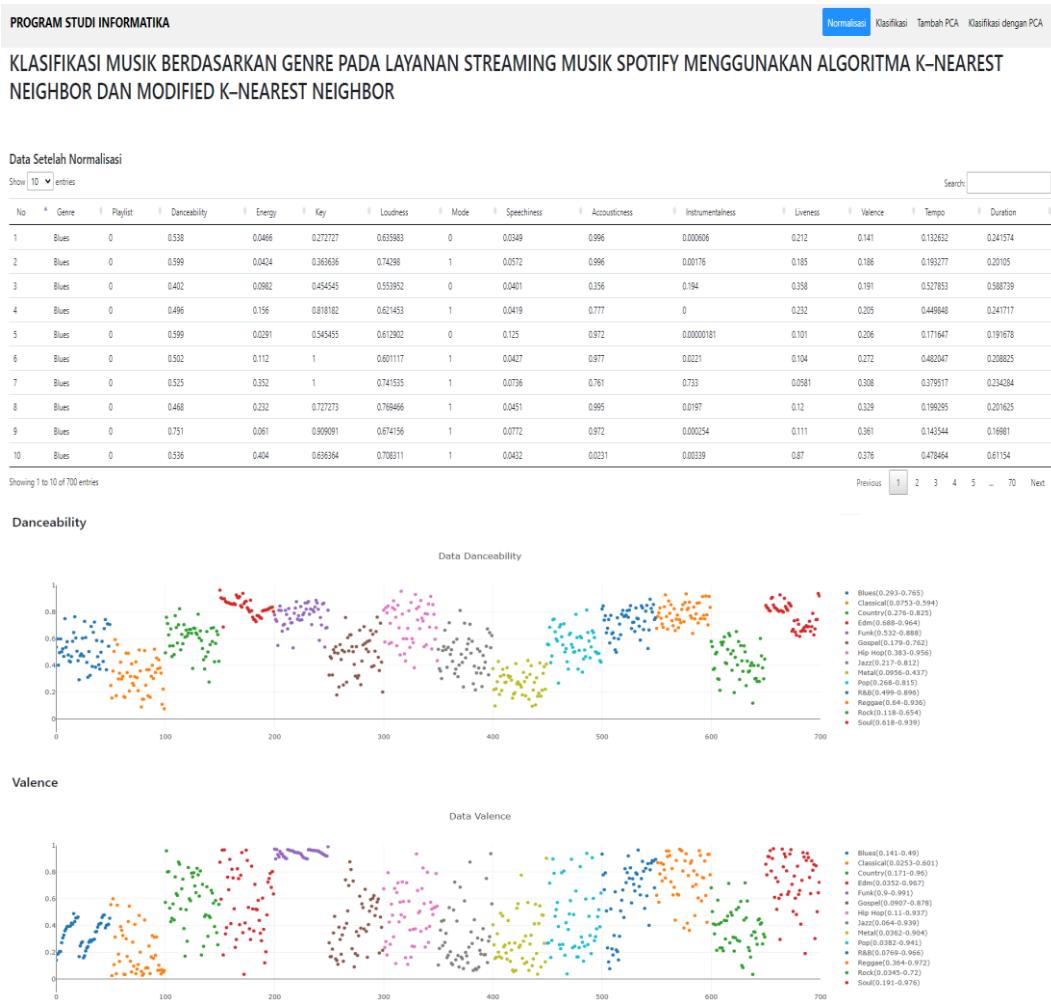
Sedangkan pengujian kedua, proses Persamaan (2.10) dan Persamaan (2.11) tersebut berada pada baris ke 192 sampai dengan 230 pada Lampiran bagian C.3. Pembobotan terdapat pada baris ke 231 sampai dengan 238 pada Lampiran bagian C.3. Selanjutnya kelas data dengan bobot tertinggi menjadi hasil kelas data dari data uji tersebut, maka data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut sesuai Persamaan (2.13), yang ditujukan pada baris ke 239 sampai dengan 251 pada Lampiran bagian C.3.

## 4.6. Implementasi Sistem

Pengguna melakukan *min–max normalization* yang nantinya akan digunakan untuk proses klasifikasi data. Form dari sebelum normalisasi dengan sebaran data pada Gambar 4.31 dan setelah normalisasi dengan sebaran data pada Gambar 4.32.



Gambar 4.31. Sebelum Normalisasi dengan sebaran data



Gambar 4.32. Sesudah Normalisasi dengan sebaran data

Implementasi Sistem lainnya bisa dilihat pada Lampiran B.3. Implementasi Sistem pada KNN dan MKNN untuk Pengujian Pertama, sedangkan Lampiran C.4. Implementasi Sistem pada PCA-KNN dan PCA-MKNN untuk Pengujian Kedua.

#### 4.7. Hasil Pengujian

Pengujian penelitian ini dilakukan dengan menghitung rata-rata akurasi tiap  $k$ -fold dari metode yang diajukan yaitu MKNN. Untuk mengetahui keefektifan metode yang diajukan, maka dibandingkan dengan metode KNN. Perbandingan dengan metode KNN, dikarenakan metode MKNN merupakan modifikasi dari metode MKNN. Selain itu, ditampilkan kelas data yang salah dikenali oleh KNN dan MKNN.

#### 4.5.1. 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 *shuffle* data terlebih dahulu, agar data menjadi independent karena data sebelumnya tersebut berurutan sesuai kelas nya. Pada Tabel 4.5 disajikan tampilan data yang sudah di *shuffle*.

Tabel 4.5. *Shuffle* Data

No	Playlist	Danceability	Energy	Key	Loudness	Mode	Speechiness	Acousticness
1	5	0.541	0.675	0.090909	0.898785	1	0.066	0.1
2	11	0.766	0.49	0.545455	0.719925	1	0.3	0.359
3	6	0.479	0.549	1	0.780909	0	0.373	0.57
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
698	7	0.574	0.409	0.363636	0.67249	0	0.0292	0.933
699	3	0.773	0.608	0.272727	0.869334	0	0.121	0.278
700	7	0.263	0.209	0.636364	0.744989	0	0.0334	0.989

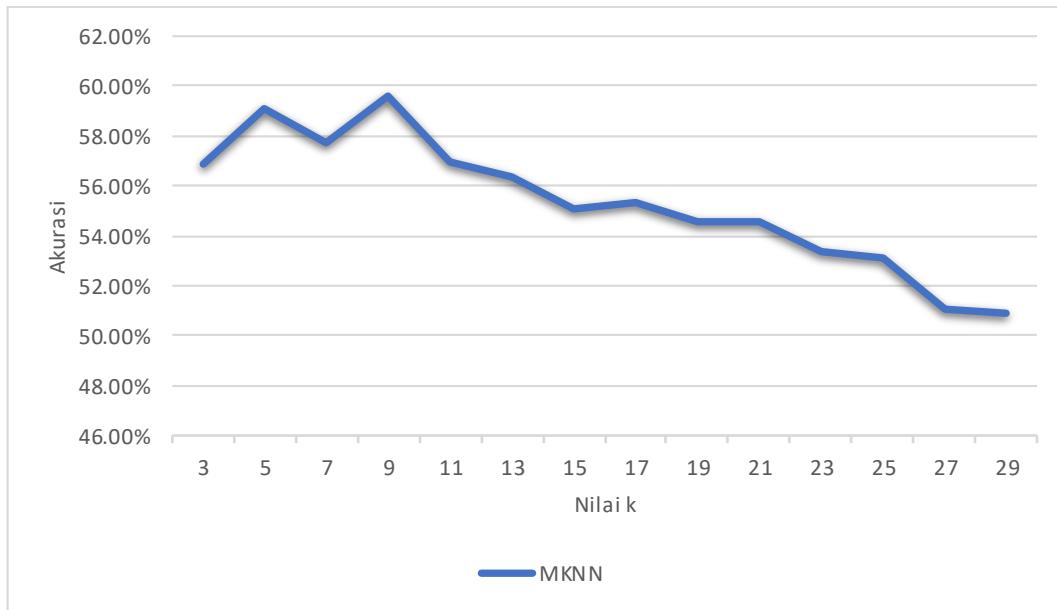
No	Playlist	Instrumentalness	Liveness	Valence	Tempo	Duration
1	5	0	0.977	0.355	0.425518	0.429394
2	11	0.0085	0.046	0.959	0.081912	0.187261
3	6	0.0239	0.127	0.576	0.783917	0.303463
-	-	-	-	-	-	-
-	-	-	-	-	-	-
698	7	0.931	0.0833	0.517	0.237856	0.349318
699	3	0	0.381	0.589	0.420986	0.196709
700	7	0.944	0.138	0.064	0.739651	0.420973

##### 4.5.1.1. Pengaruh Perubahan Nilai $k$ terhadap Akurasi MKNN

Nilai  $k$  merupakan bilangan asli (bilangan bulat positif yang bukan 0). Bilangan asli diperlukan karena metode MKNN memerlukan minimal 1 data latih

terdekat. Selain itu, nilai  $k$  juga akan menggunakan bilangan ganjil agar meningkatkan kecepatan algoritma dengan menghindari pengklasifikasi genap untuk menghindari kemungkinan dua kelas berbeda memiliki jumlah suara yang sama (Hassanat, 2004). Nilai  $k$  yang digunakan pada penelitian ini yaitu bilangan ganjil kisaran 3 sampai dengan 29 karena pemilihan nilai  $k$  tersebut berdasarkan hasil akurasi yang didapatkan. Pada Gambar 4.33 menunjukkan hasil akurasi mengalami fluktuasi (tren grafik turun, naik dan turun) saat  $k$  bernilai 11 sampai dengan 29 rata-rata akurasi cenderung mengalami tren turun dan naik. Sehingga rentang nilai  $k$  tersebut telah mewakili bahwa tidak ada perbaikan nilai akurasi yang terjadi. Oleh karena itu, pengujian hanya dilakukan hingga  $k$  bernilai 29.  $k$  bernilai 1 tidak digunakan dalam pengujian ini, karena nilai  $k$  tersebut tidak cocok digunakan pada metode MKNN. Ketidakcocokan tersebut dikarenakan hanya 1 data latih terdekat dari data uji yang dicari, maka tidak diperlukan proses pembobotan atau sama saja dengan metode KNN.

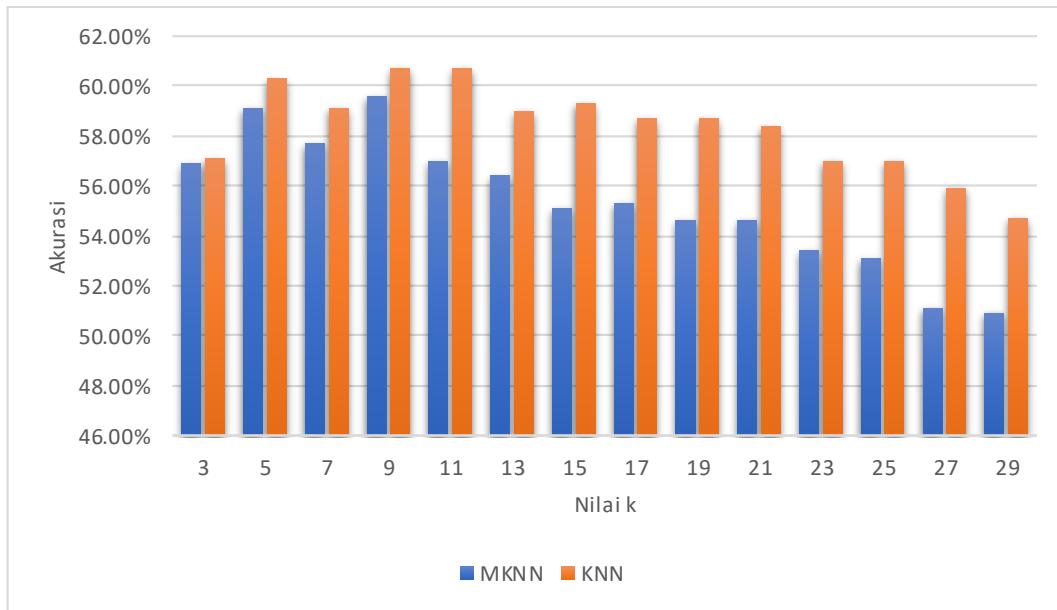
Beberapa nilai  $k$  yang digunakan pada metode MKNN menghasilkan nilai akurasi yang berbeda. Pada Gambar 4.33 menampilkan akurasi MKNN dengan data uji yaitu *genre*. Nilai akurasi tertinggi terjadi ketika nilai  $k = 9$ , dimana nilai akurasi yang diperoleh yaitu 59.6%. Nilai akurasi terendah terjadi pada saat nilai  $k = 29$ , dimana nilai akurasi yang diperoleh yaitu 50.9%. Selanjutnya nilai  $k = 19$  dan  $k = 21$  memiliki nilai akurasi yang sama yaitu 54.6%. Pada Gambar 4.33 juga menunjukkan terjadi kecenderungan kenaikan nilai rata-rata akurasi dari  $k = 3$  dan rata-rata akurasi tertinggi saat  $k = 5$  dengan nilai akurasi yang diperoleh yaitu 59.1%. Kemudian terjadi penurunan rata-rata akurasi ketika  $k = 7$ . Perbaikan tingkat akurasi kembali terjadi ketika  $k$  bernilai 9 dengan nilai akurasi yang diperoleh yaitu 59.6% yang menjadikan rata-rata akurasi tertinggi dari MKNN.



Gambar 4.33. Akurasi MKNN

#### 4.5.1.2. Perbandingan Akurasi pada perubahan nilai $k$ MKNN dengan KNN

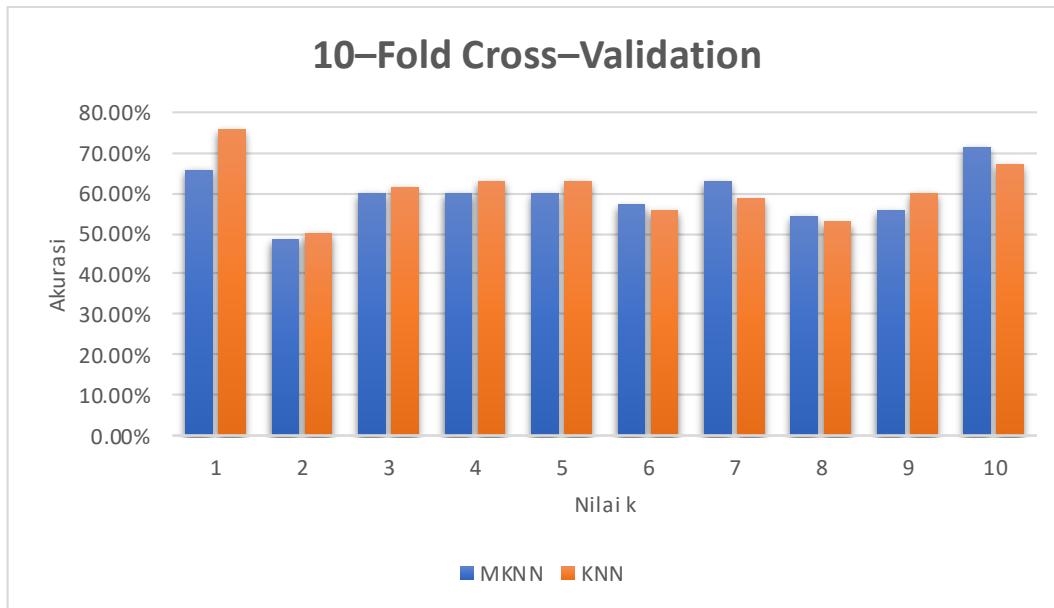
Akurasi klasifikasi data uji *genre* dari MKNN lebih rendah dibandingkan KNN pada semua nilai  $k$ . Akurasi MKNN memiliki akurasi lebih rendah dibandingkan dengan KNN pada saat nilai  $k = 3$  sampai dengan  $k = 29$ . Dari keseluruhan nilai  $k$  pada data uji klasifikasi *genre*, metode MKNN memiliki 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  $9$ . Namun, pada saat nilai  $k = 11$  metode MKNN mengalami penurunan akurasi menjadi  $57\%$ . Sehingga ketika nilai  $k = 9$  ( $k$  yang paling optimal) metode KNN meningkatkan nilai akurasi dari metode MKNN pada saat nilai  $k = 9$  sebanyak  $1.1\%$ . Pada Gambar 4.34 disajikan hasil pengujian perbandingan akurasi MKNN dengan KNN pada klasifikasi *genre*.



Gambar 4.34. Perbandingan Akurasi MKNN dengan KNN pada Klasifikasi *Genre*

#### 4.5.1.3. Perbandingan Akurasi *10–Fold Cross–Validation* MKNN dengan KNN pada $k = 9$

Akurasi pada partisi *10-fold cross-validation* pada metode MKNN lebih rendah dibandingkan dengan metode KNN. Partisi *10-fold cross-validation* diambil dari kedua metode MKNN dan KNN saat nilai  $k = 9$ . Pada saat nilai  $k$ -fold iterasi ke-1 dengan nilai akurasi 75.7% merupakan akurasi tertinggi pada metode KNN, sedangkan saat nilai  $k$ -fold iterasi ke-1 dengan nilai akurasi 65.7% merupakan akurasi pada metode MKNN. Selanjutnya pada saat nilai  $k$ -fold iterasi ke-10 dengan nilai akurasi 71.4% merupakan akurasi tertinggi pada metode MKNN, sedangkan saat nilai  $k$ -fold iterasi ke-10 dengan nilai akurasi 67.1% merupakan akurasi pada metode KNN. Pada Gambar 4.35 disajikan hasil perbandingan akurasi *10-fold cross-validation* MKNN dengan KNN pada  $k = 9$ .



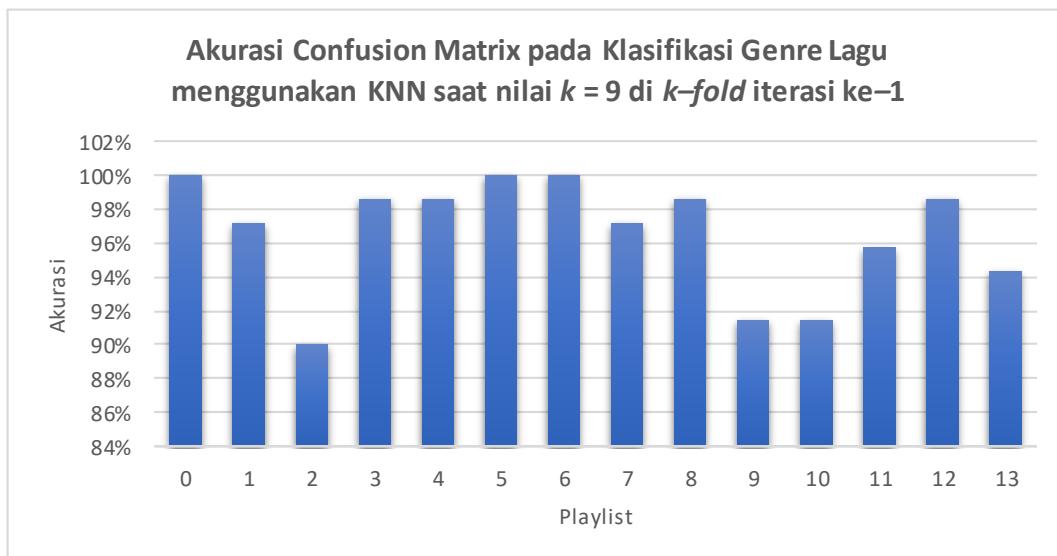
Gambar 4.35. Perbandingan Akurasi 10-Fold Cross-Validation MKNN dengan KNN pada  $k = 9$

#### 4.5.1.4. Hasil Evaluasi *Confusion Matrix* Pada Klasifikasi *Genre Lagu* menggunakan KNN saat nilai $k = 9$ di $k$ -*fold* iterasi ke-1

Hasil evaluasi pada klasifikasi *genre* lagu menggunakan metode KNN dengan *confusion matrix* untuk mengukur seberapa tepat klasifikasi terhadap kelas dalam pengujian data uji dengan data latih. Proses ini dilakukan dengan *confusion matrix* menggunakan perhitungan akurasi pada Persamaan (2.7). Pada Tabel 4.6 disajikan hasil evaluasi *Confusion Matrix* pada Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -*fold* iterasi ke-1:

Tabel 4.6. Hasil Evaluasi *Confusion Matrix* pada Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -*fold* iterasi ke-1

6	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	2	1	1	0	0	0
10	0	0	1	1	0	0	0	0	0	1	1	1	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	4	1	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
13	0	0	2	0	0	0	0	0	0	0	1	0	0	0	2
TP	5	4	1	5	5	5	5	4	5	2	1	4	5	2	
FP	0	1	3	1	1	0	0	1	1	3	2	2	1	1	
TN	65	64	62	64	64	65	65	64	64	62	63	63	64	64	
FN	0	1	4	0	0	0	0	1	0	3	4	1	0	3	



Gambar 4.36. Hasil Akurasi Klasifikasi *Genre Lagu* menggunakan KNN pada *Confusion Matrix* saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1

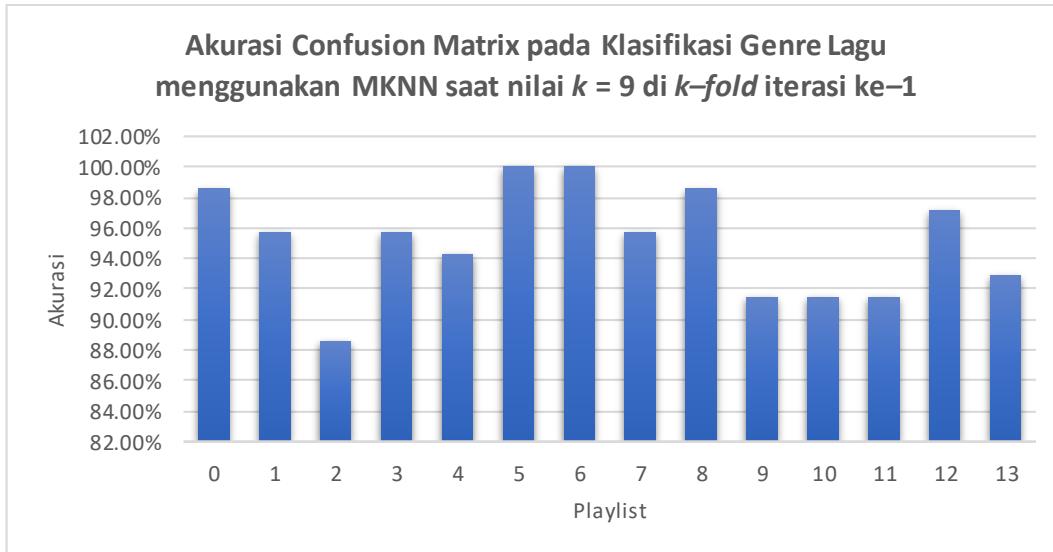
Dari perhitungan akurasi pada Persamaan (2.7). Pada Gambar 4.36 menunjukkan akurasi tiap kelas *genre* dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) dan *False Negative* (FN). Pada *playlist* label 1 *Classical*, label 5 *Gospel* dan label 6 *Hip Hop* mendapatkan akurasi sebesar 100%, disusul *playlist* label 3 *EDM*, label 4 *Funk*, label 8 *Metal* dan label 12 *Rock* mendapatkan akurasi yang sama sebesar 99%, sedangkan *playlist* label 2 *Country* mendapatkan akurasi terendah yaitu 90%. Untuk tabel *confusion matrix* lainnya bisa dilihat pada Lampiran D.3.

#### 4.5.1.5. Hasil Evaluasi *Confusion Matrix* Pada Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-1

Hasil evaluasi pada klasifikasi *genre lagu* menggunakan metode KNN dengan *confusion matrix* untuk mengukur seberapa tepat klasifikasi terhadap kelas dalam pengujian data uji dengan data latih. Proses ini dilakukan dengan *confusion matrix* menggunakan perhitungan akurasi pada Persamaan (2.7). Pada Tabel 4.7 disajikan hasil evaluasi *Confusion Matrix* pada Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1:

Tabel 4.7. Hasil Evaluasi *Confusion Matrix* pada Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	3	0	0	0	0	0	0	2	0	0	0	0	0	0
	2	1	0	0	0	1	0	0	0	0	2	0	0	0	0	1
	3	0	0	0	3	0	0	0	0	0	0	0	0	1	1	0
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	1	2	1	1	0	0	0
	10	0	0	1	1	0	0	0	0	0	1	0	2	0	0	0
	11	0	0	0	0	1	0	0	0	0	0	0	3	1	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	13	0	0	2	0	2	0	0	0	0	0	0	0	0	0	1
TP		5	3	0	3	5	5	5	4	5	2	0	3	5	1	
FP		1	1	3	1	4	0	0	2	1	3	1	4	2	1	
TN		64	64	62	64	61	65	65	63	64	62	64	61	63	64	
FN		0	2	5	2	0	0	0	1	0	3	5	2	0	4	

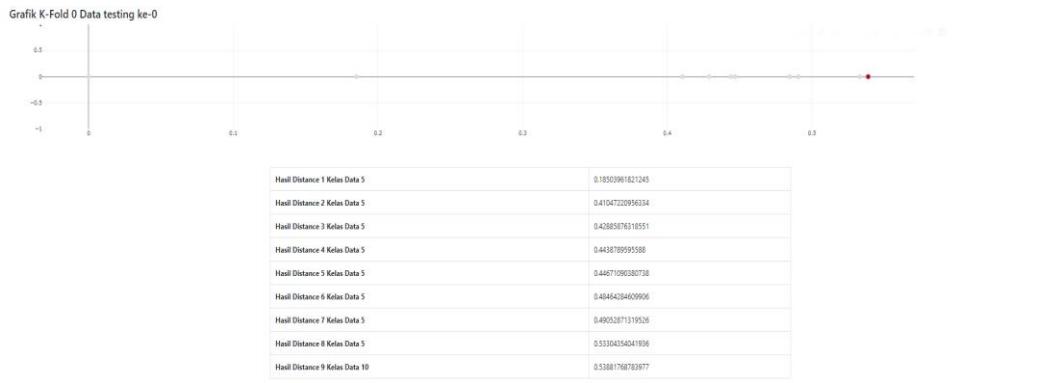


Gambar 4.37. Akurasi Klasifikasi *Genre Lagu* menggunakan MKNN pada *Confusion Matrix* saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1

Dari perhitungan akurasi pada Persamaan (2.7). Pada Gambar 4.37 menunjukkan akurasi tiap kelas *genre* dari *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) dan *False Negative* (FN). Pada *playlist* label 1 *Classical*, label 5 *Gospel* dan label 6 *Hip Hop* mendapatkan akurasi sebesar 100%, disusul *playlist* label 3 *EDM*, label 4 *Funk*, label 8 *Metal* dan label 12 *Rock* mendapatkan akurasi yang sama sebesar 99%, sedangkan *playlist* label 2 *Country* mendapatkan akurasi terendah yaitu 90%. Untuk tabel *confusion matrix* lainnya bisa dilihat pada Lampiran D.4.

#### 4.5.1.6. Identifikasi Pemilihan Kelas pada KNN saat nilai $k = 9$ di $k$ -fold iterasi ke-1

Dalam pemilihan kelas yang telah dibangun terdapat beberapa hasil data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian pada Tabel 4.6. Diperoleh hasil pemilihan kelas yang benar diklasifikasi maupun yang salah diklasifikasi pada metode KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1 sebagai berikut:



Gambar 4.38. Perhitungan *euclidean distance* KNN sejumlah  $k = 9$  pada data *testing* ke-0

Pada Gambar 4.38 menunjukkan metode KNN melakukan perhitungan *euclidean distance* menggunakan cara pada Persamaan (2.8). Selanjutnya nilai *euclidean distance* akan diurutkan dari terendah hingga terbesar sejumlah  $k$  buah. Pada saat nilai  $k = 9$  maka dipilih *euclidean distance* sejumlah 9 nilai *euclidean distance*. Diperoleh hasil sebagai berikut:

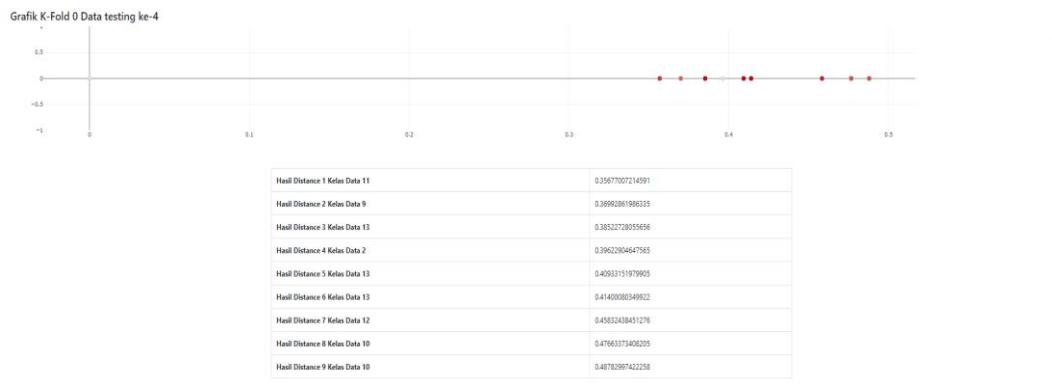
<b>Hasil Distance 1 Kelas Data 5</b>	: 0.18503961821245
<b>Hasil Distance 2 Kelas Data 5</b>	: 0.41047220956334
<b>Hasil Distance 3 Kelas Data 5</b>	: 0.42885876318551
<b>Hasil Distance 4 Kelas Data 5</b>	: 0.4438789595588
<b>Hasil Distance 5 Kelas Data 5</b>	: 0.44671090380738
<b>Hasil Distance 6 Kelas Data 5</b>	: 0.48464284609906
<b>Hasil Distance 7 Kelas Data 5</b>	: 0.49052871319526
<b>Hasil Distance 8 Kelas Data 5</b>	: 0.53304354041936
<b>Hasil Distance 9 Kelas Data 10</b>	: 0.53881768783977

Kelas Data Uji	5
Kelas Data Hasil Klasifikasi	5

Gambar 4.39. Kelas Data Hasil Klasifikasi KNN pada data *testing* ke-0 yang terpilih benar

Selanjutnya memilih kelas data yang terbanyak pertama yang merupakan kelas data dari data uji tersebut, maka data uji dapat diketahui kelas datanya atau

*genre* dari lagu tersebut. Pada Gambar 4.39 menunjukkan kelas label 5 yang sebagai kelas data uji pada data *testing* ke-0 memilih kelas data terbanyak pertama yang akan dipilih sebagai kelas data hasil klasifikasi dan kelas data hasil klasifikasi KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1 pada data *testing* ke-0 yang terpilih adalah label 5 yaitu benar, sedangkan pemilihan label *genre* yang salah pada klasifikasi KNN sebagai berikut:



Gambar 4.40. Perhitungan *euclidean distance* KNN sejumlah  $k = 9$  pada data *testing* ke-1

Dilihat pada Gambar 4.40 menunjukkan metode KNN melakukan perhitungan *euclidean distance* menggunakan cara pada Persamaan (2.8). Selanjutnya nilai *euclidean distance* akan diurutkan dari terendah hingga terbesar sejumlah  $k$  buah. Pada saat nilai  $k = 9$  maka dipilih *euclidean distance* sejumlah 9 nilai *euclidean distance*. Diperoleh hasil sebagai berikut:

<b>Hasil Distance 1 Kelas Data 11</b>	: 0.35677007214591
<b>Hasil Distance 2 Kelas Data 9</b>	: 0.36992861986335
<b>Hasil Distance 3 Kelas Data 13</b>	: 0.38522728055656
<b>Hasil Distance 4 Kelas Data 2</b>	: 0.39622904647565
<b>Hasil Distance 5 Kelas Data 13</b>	: 0.40933151979905
<b>Hasil Distance 6 Kelas Data 13</b>	: 0.41400080349922
<b>Hasil Distance 7 Kelas Data 12</b>	: 0.45832438451276
<b>Hasil Distance 8 Kelas Data 10</b>	: 0.47663373408205
<b>Hasil Distance 9 Kelas Data 10</b>	: 0.48782997422258

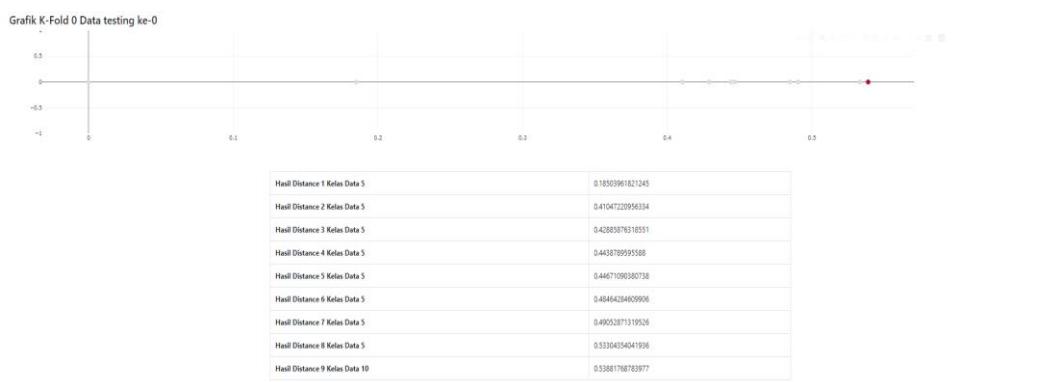
Kelas Data Uji	2
Kelas Data Hasil Klasifikasi	13

Gambar 4.41. Kelas Data Hasil Klasifikasi KNN pada data *testing* ke-4 yang terpilih salah

Selanjutnya memilih kelas data yang terbanyak pertama yang merupakan kelas data dari data uji tersebut, maka data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut. Pada Gambar 4.41 menunjukkan kelas label 2 yang sebagai kelas data uji pada data *testing* ke-1 memilih kelas data terbanyak pertama yang akan dipilih sebagai kelas data hasil klasifikasi dan kelas data hasil klasifikasi KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1 pada data *testing* ke-4 yang terpilih adalah label 13 yaitu salah.

#### 4.5.1.7. Identifikasi Pemilihan Kelas pada MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-1

Dalam model sistem yang telah dibangun, terdapat beberapa hasil data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian pada Tabel 4.7. Diperoleh hasil pemilihan kelas yang benar diklasifikasi maupun yang salah diklasifikasi MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-1 sebagai berikut:



Gambar 4.42. Perhitungan *euclidean distance* MKNN sejumlah  $k = 9$  pada data *testing* ke-0

Pada Gambar 4.42 menunjukkan metode MKNN melakukan perhitungan *euclidean distance* menggunakan cara pada Persamaan (2.8). Selanjutnya nilai *euclidean distance* akan diurutkan dari terendah hingga terbesar sejumlah  $k$  buah. Pada saat nilai  $k = 9$  maka dipilih sejumlah 9 nilai *euclidean distance*. Diperoleh hasil sebagai berikut:

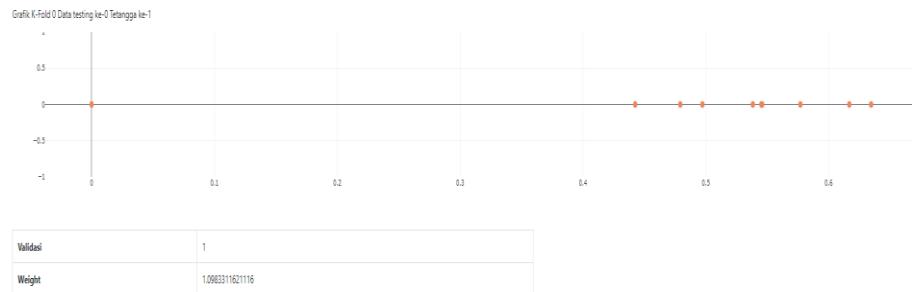
<b>Hasil Distance 1 Kelas Data 5</b>	:	0.18503961821245
<b>Hasil Distance 2 Kelas Data 5</b>	:	0.41047220956334
<b>Hasil Distance 3 Kelas Data 5</b>	:	0.42885876318551
<b>Hasil Distance 4 Kelas Data 5</b>	:	0.4438789595588
<b>Hasil Distance 5 Kelas Data 5</b>	:	0.44671090380738
<b>Hasil Distance 6 Kelas Data 5</b>	:	0.48464284609906
<b>Hasil Distance 7 Kelas Data 5</b>	:	0.49052871319526
<b>Hasil Distance 8 Kelas Data 5</b>	:	0.53304354041936
<b>Hasil Distance 9 Kelas Data 10</b>	:	0.53881768783977

Selanjutnya tiap *euclidean distance* terendah pertama dan seterusnya akan dihitung nilai validasi dengan cara pada Persamaan (2.9) dan Persamaan (2.10) dengan tetangga data latih lainnya sejumlah  $k = 5$ . Jika data latih sama dengan data latih tetangganya maka bernilai 1, tetapi jika tidak sama dengan data latih tetangganya maka bernilai 0 dengan cara pada Persamaan (2.10). Selanjutnya jumlah seluruh validitas dibagi dengan sejumlah  $k$  buah dengan cara pada Persamaan (2.9). Dari validasi tersebut didapatkan bobot dengan cara pada Persamaan (2.11) pada masing-masing kelas datanya. Kemudian pemilihan bobot akan dilakukan dengan cara pada Persamaan (2.12) dengan memilih bobot yang tertinggi yang merupakan kelas data dari data uji tersebut, maka Data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut.



Gambar 4.43. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-0

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-0 pada Gambar 4.43 yaitu nilai validasi adalah 0.88888888888889 dan nilai bobot adalah 1.2975729655.



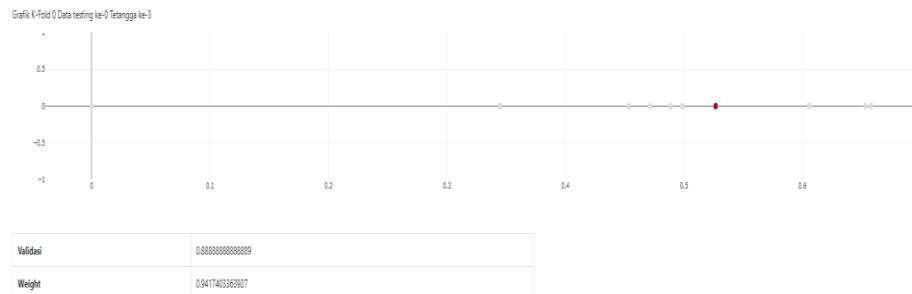
Gambar 4.44. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-1

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-1 pada Gambar 4.44 yaitu nilai validasi adalah 1 dan nilai bobot adalah 1.0983311621116.



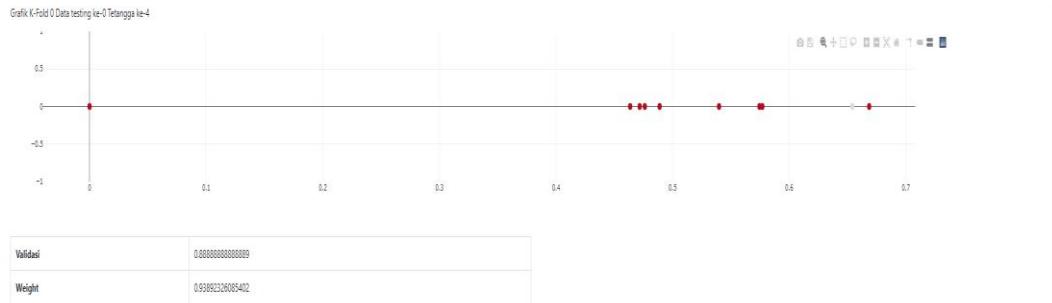
Gambar 4.45. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-2

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-2 pada Gambar 4.45 yaitu nilai validasi adalah 1 dan nilai bobot adalah 1.076589939864.



Gambar 4.46. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-3

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-3 pada Gambar 4.46 yaitu nilai validasi adalah 0.88888888888889 dan nilai bobot adalah 0.9417403363927.



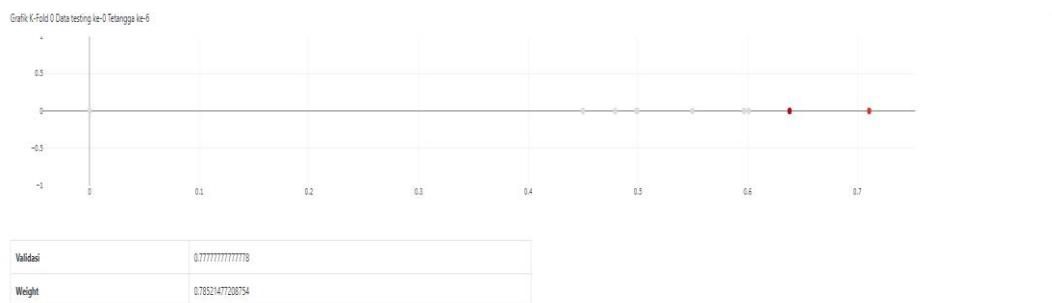
Gambar 4.47. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-4

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-4 pada Gambar 4.47 yaitu nilai validasi adalah 0.88888888888889 dan nilai bobot adalah 0.93892326085402.



Gambar 4.48. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-5

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-5 pada Gambar 4.48 yaitu nilai validasi adalah 0.88888888888889 dan nilai bobot adalah 0.90275259949379.



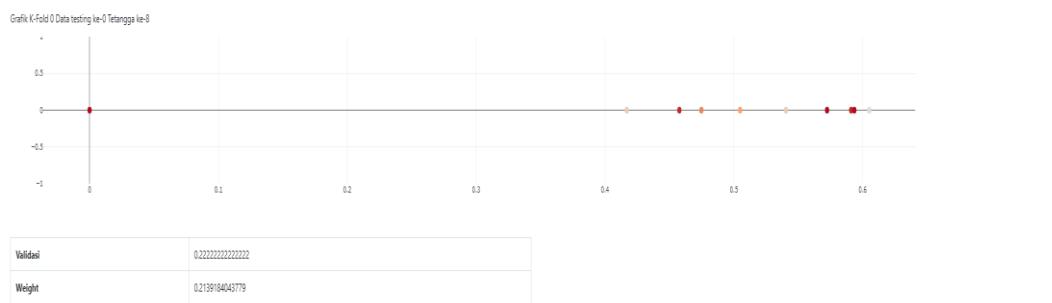
Gambar 4.49. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-6

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-6 pada Gambar 4.49 yaitu nilai validasi adalah 0.77777777777778 dan nilai bobot adalah 0.78521477208754.



Gambar 4.50. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-7

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-7 pada Gambar 4.50 yaitu nilai validasi adalah 1 dan nilai bobot adalah 0.96801340976785.



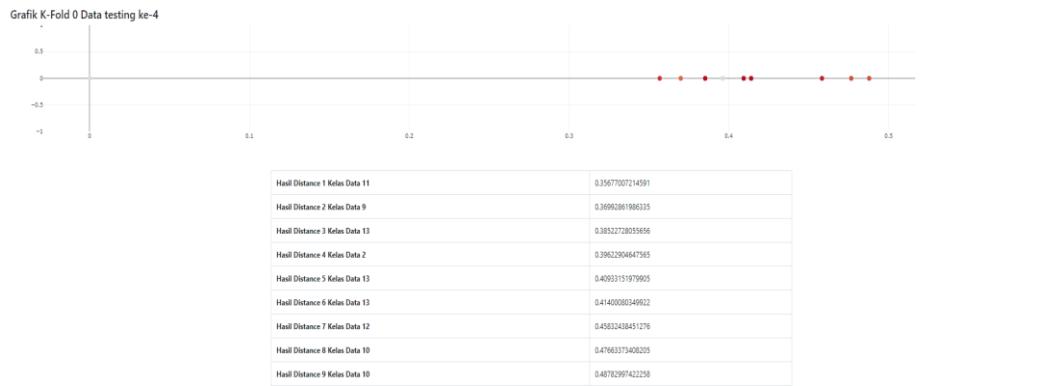
Gambar 4.51. Menghitung validasi dan bobot data *testing* ke-0 tetangga ke-8

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-0 dengan tetangga ke-8 pada Gambar 4.51 yaitu nilai validasi adalah 0.2222222222222222 dan nilai bobot adalah 0.2139184043779.

Kelas Data Uji	5
Kelas Data Hasil Klasifikasi	5

Gambar 4.52. Kelas Data Hasil Klasifikasi MKNN pada data *testing* ke-0 yang terpilih benar

Dari rumus Persamaan (2.12) pada Gambar 4.52 menunjukkan kelas label 5 yang sebagai kelas data uji pada data *testing* ke-0 memilih bobot yang tertinggi yang akan terpilih sebagai kelas data hasil klasifikasi. Pada klasifikasi MKNN saat nilai  $k = 9$  dan  $k$ -*fold* iterasi ke-1 dan data *testing* ke-0 menghasilkan kelas data hasil klasifikasi yang terpilih adalah label 5 yaitu benar, sedangkan pemilihan label *genre* yang salah pada MKNN sebagai berikut:



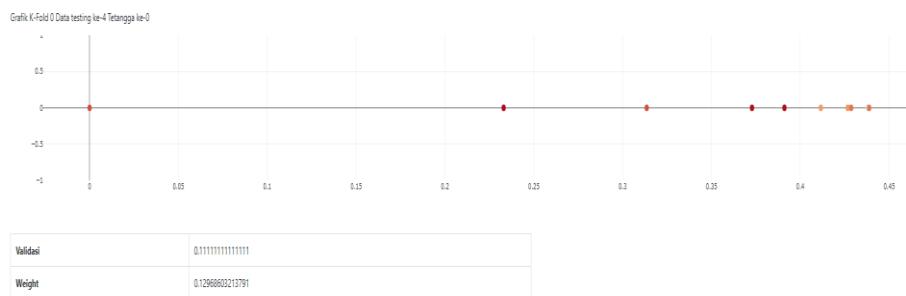
Gambar 4.53. Perhitungan *euclidean distance* MKNN sejumlah  $k = 9$  pada data *testing* ke-4

Dilihat pada Gambar 4.53, metode MKNN melakukan perhitungan jarak *euclidean distance* terlebih dahulu menggunakan cara pada Persamaan (2.9). Setelah itu, mengurutkan jarak *euclidean distance*, dari terendah hingga terbesar. Selanjutnya menghitung validasi dan bobot tiap tetangga nya sejumlah  $k$ . Pada  $k = 9$  hasil *distance* terpilih sejumlah  $k = 9$  sebagai berikut:

- Hasil Distance 1 Kelas Data 11 : 0.35677007214591**
- Hasil Distance 2 Kelas Data 9 : 0.36992861986335**
- Hasil Distance 3 Kelas Data 13 : 0.38522728055656**
- Hasil Distance 4 Kelas Data 2 : 0.39622904647565**
- Hasil Distance 5 Kelas Data 13 : 0.40933151979905**
- Hasil Distance 6 Kelas Data 13 : 0.41400080349922**
- Hasil Distance 7 Kelas Data 12 : 0.45832438451276**
- Hasil Distance 8 Kelas Data 10 : 0.47663373408205**
- Hasil Distance 9 Kelas Data 10 : 0.48782997422258**

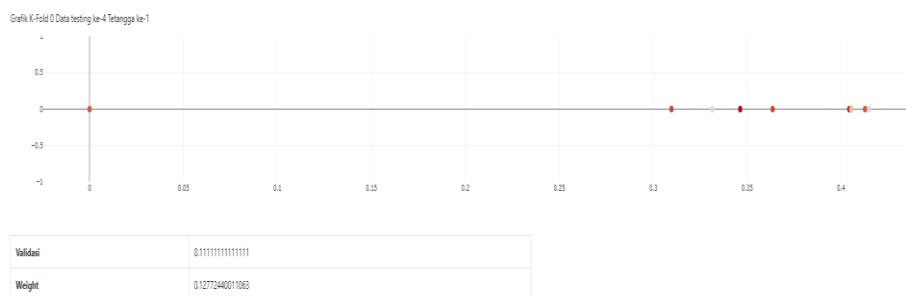
Selanjutnya tiap *euclidean distance* terendah pertama dan seterusnya akan dihitung nilai validasi dengan cara pada Persamaan (2.9) dan Persamaan (2.10)

dengan tetangga data latih lainnya sejumlah  $k = 9$ . Jika data latih sama dengan data latih tetangganya maka bernilai 1, tetapi jika tidak sama dengan data latih tetangganya maka bernilai 0 dengan cara pada Persamaan (2.10). Selanjutnya jumlah seluruh validitas dibagi dengan sejumlah  $k$  buah dengan cara pada Persamaan (2.9). Dari validasi tersebut didapatkan bobot dengan cara pada Persamaan (2.11) pada masing-masing kelas datanya. Kemudian pemilihan bobot akan dilakukan dengan cara pada Persamaan (2.12) dengan memilih bobot yang tertinggi yang merupakan kelas data dari data uji tersebut, maka Data uji dapat diketahui kelas datanya atau *genre* dari lagu tersebut.



Gambar 4.54. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-0

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-0 pada Gambar 4.54 yaitu nilai validasi yang didapatkan adalah 0.1111111111111111 dan nilai bobot yang didapatkan adalah 0.12968603213791.



Gambar 4.55. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-1

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-1 pada Gambar 4.55 yaitu

nilai validasi yang didapatkan adalah 0.11111111111111 dan nilai bobot yang didapatkan adalah 0.12772440011063.



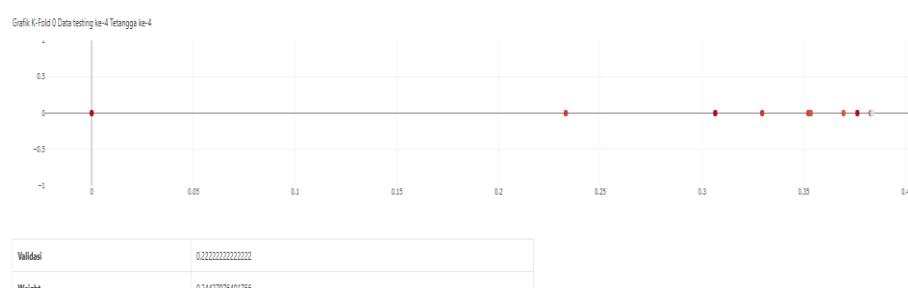
Gambar 4.56. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-2

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-2 pada Gambar 4.56 yaitu nilai validasi yang didapatkan adalah 0.33333333333333 dan bobot yang didapatkan adalah 0.37655113060203.



Gambar 4.57. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-3

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-3 pada Gambar 4.57 yaitu nilai validasi yang didapatkan adalah 0.11111111111111 dan yang didapatkan adalah 0.12397624418451.



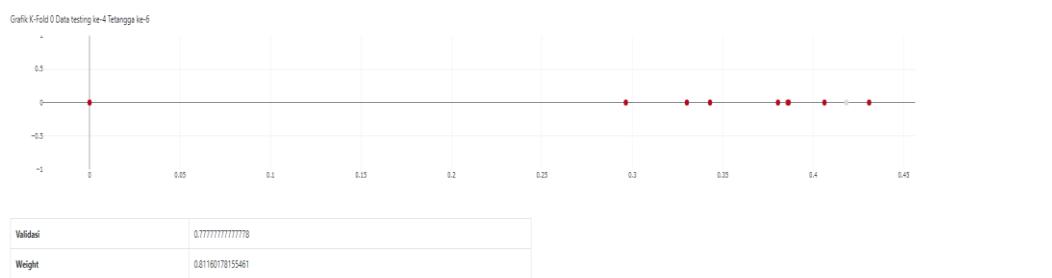
Gambar 4.58. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-4

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-4 pada Gambar 4.58 yaitu nilai validasi yang didapatkan adalah 0.2222222222222222 dan bobot yang didapatkan adalah 0.24437976401756.



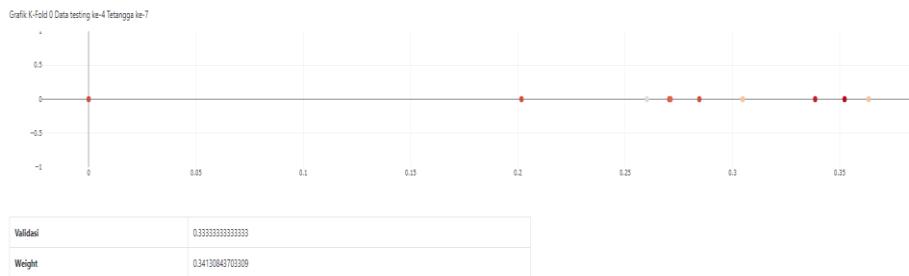
Gambar 4.59. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-5

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-5 pada Gambar 4.59 yaitu nilai validasi yang didapatkan adalah 0.2222222222222222 dan nilai bobot yang didapatkan adalah 0.24313132042275.



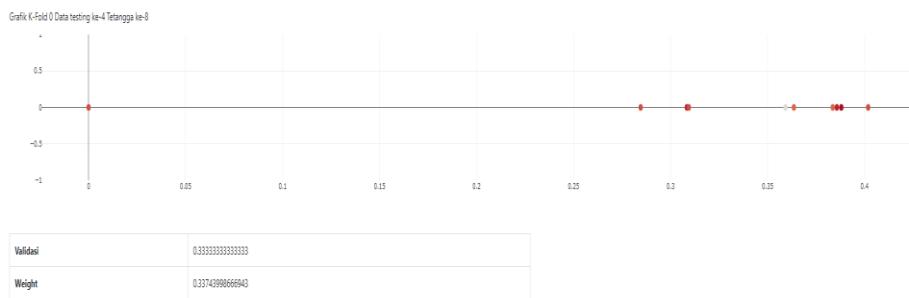
Gambar 4.60. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-6

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-6 pada Gambar 4.60 yaitu nilai validasi yang didapatkan adalah 0.7777777777777778 dan bobot yang didapatkan adalah 0.81160178155461.



Gambar 4.61. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-7

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-7 pada Gambar 4.61 yaitu nilai validasi yang didapatkan adalah 0.33333333333333 dan yang didapatkan adalah 0.34130843703309.



Gambar 4.62. Menghitung validasi dan bobot data *testing* ke-4 tetangga ke-8

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data *testing* ke-4 dengan tetangga ke-8 pada Gambar 4.62 yaitu nilai validasi yang didapatkan adalah 0.33333333333333 dan bobot yang didapatkan adalah 0.33743998666943.

Kelas Data Uji	2
Kelas Data Hasil Klasifikasi	13

Gambar 4.63. Kelas Data Hasil Klasifikasi MKNN pada data *testing* ke-4 yang terpilih salah

Dari rumus Persamaan (2.12) pada Gambar 4.63 menunjukkan kelas label 2 yang sebagai kelas data uji pada data *testing* ke-4 memilih bobot yang tertinggi yang akan terpilih sebagai kelas data hasil klasifikasi. Pada klasifikasi MKNN saat

nilai  $k = 9$  dan  $k$ -fold iterasi ke-1 dan data *testing* ke-4 menghasilkan kelas data hasil klasifikasi yang terpilih adalah label 13 yaitu salah.

#### 4.5.2. 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  $\lambda$  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). Dari Persamaan diatas, hasil reduksi menggunakan PCA pada Lampiran C.1. Berikut adalah nilai eigen value dan persentase tiap fitur pada tabel 4.6.

Tabel 4.6. Nilai EigenValue dan Persentase tiap Fitur

Fitur	Eigenvalue	ComputePercentageExplained
Danceability	17.324267064992977	0.31452914321125314
Energy	13.89250122875269	0.5667531762481529
Key	6.244412386600236	0.6801230331342691
Loudness	5.732648412121658	0.7842016060765816
Mode	3.485010144691698	0.84747338841405
Speechiness	2.689476677515101	0.8963019357858931
Accousticness	1.9947499148994556	0.9325174350794733
Instrumentalness	1.297703111163251	0.95607776500082
Liveness	0.9710438097208153	0.9737074619078391
Valance	0.6763196747731051	0.9859863217894871
Tempo	0.5146757735253626	0.9953304706340654
Duration	0.2571977050436473	1

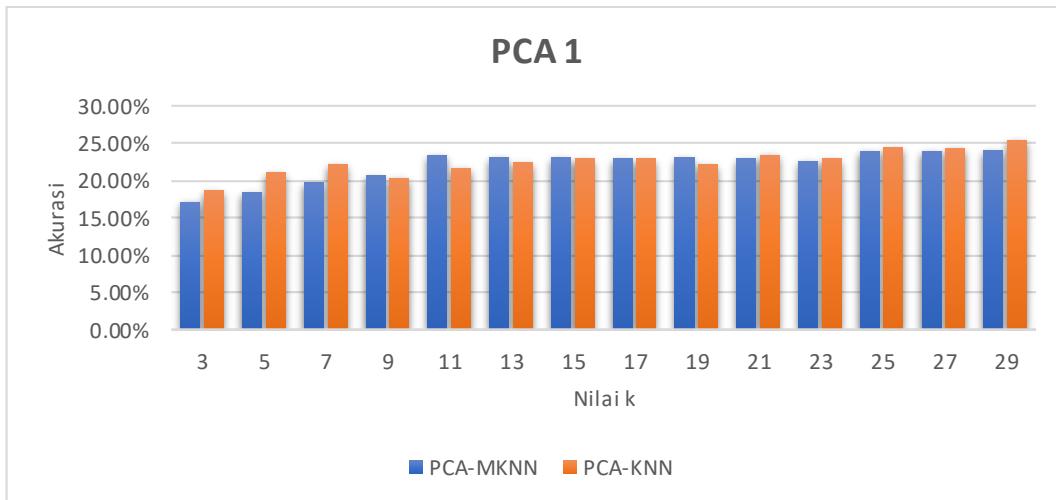
Proses KNN dan MKNN dengan menggunakan PCA pengurangan jumlah input dilakukan dimulai dari komponen dengan eigen value paling terbesar, maka jumlah variabel fitur baru yang akan diuji yaitu 1 sampai dengan 11 fitur,

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.

#### **4.5.2.1. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 1 fitur baru**

Dari keseluruhan nilai  $k$  dengan 1 fitur baru, PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA–KNN. Pada Gambar 4.64 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata–rata akurasi metode PCA–MKNN tidak melebihi akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 29$  dengan nilai akurasi 25.3% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN memiliki nilai akurasi tertinggi sebesar 24%, pada nilai  $k$  adalah 29. Sehingga ketika nilai  $k = 29$  ( $k$  yang paling optimal) metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 29$  sebanyak 1.3%. Pada Gambar 4.64 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 1 fitur baru.



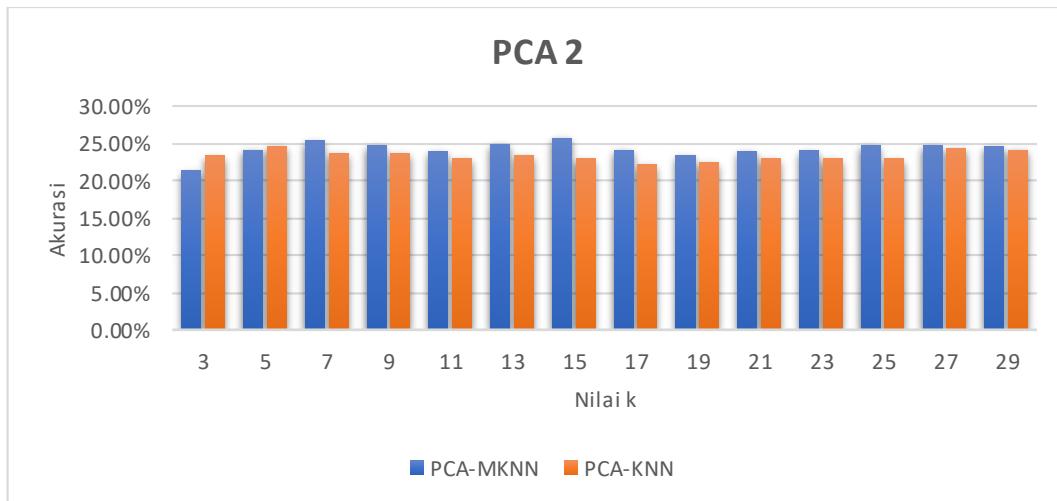
Gambar 4.64. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 1 fitur baru

#### 4.5.2.2. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 2 fitur baru

Dari keseluruhan nilai  $k$  dengan 2 fitur baru, PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan PCA–KNN. Pada Gambar 4.65 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata–rata akurasi metode PCA–MKNN meningkatkan akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 5$  dengan nilai akurasi 24.6% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN memiliki nilai akurasi sebesar 24.1%, pada nilai  $k$  adalah 5. Sehingga ketika nilai  $k = 5$  metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 5$  sebanyak 0.5%. Kemudian metode PCA–MKNN meningkatkan akurasi pada nilai  $k = 15$  dengan nilai akurasi 25.7% merupakan akurasi tertinggi pada metode PCA–MKNN, sedangkan pada metode PCA–KNN mengalami penurunan akurasi menjadi 22.9% pada nilai  $k$  adalah 15. Sehingga ketika nilai  $k =$

15 ( $k$  yang paling optimal) metode PCA–MKNN meningkatkan nilai akurasi dari metode PCA–KNN pada saat nilai  $k = 15$  sebanyak 2.8%. Pada Gambar 4.65 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 2 fitur baru.



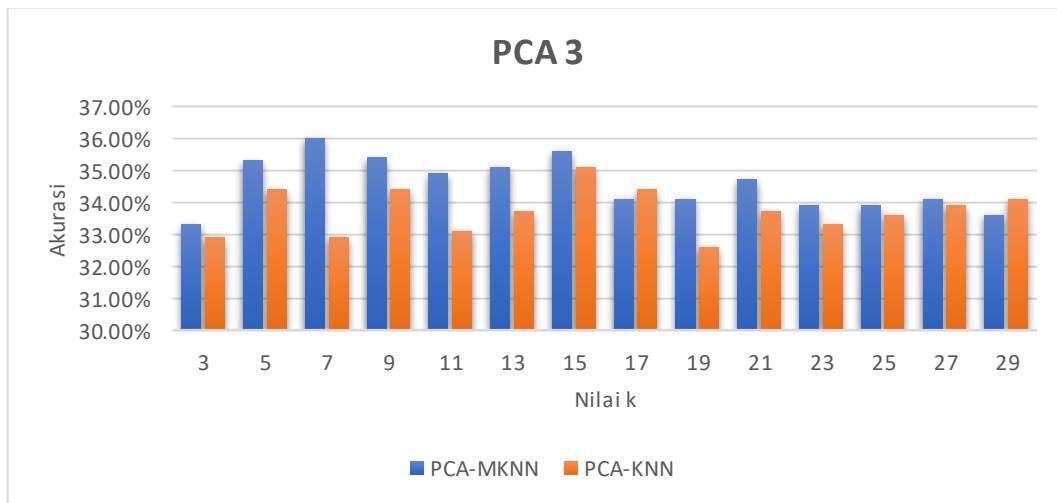
Gambar 4.65. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 2 fitur baru

#### 4.5.2.3. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 3 fitur baru

Dari keseluruhan nilai  $k$  dengan 3 fitur baru, PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan PCA–KNN. Pada Gambar 4.66 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata–rata akurasi metode PCA–MKNN meningkatkan akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan metode PCA–KNN. Saat nilai  $k = 3$  sampai  $k = 7$  nilai akurasi PCA–MKNN cenderung meningkat dibandingkan PCA–KNN. Pada nilai  $k = 7$  dengan nilai akurasi 36% merupakan akurasi tertinggi pada metode PCA–MKNN, sedangkan pada metode PCA–KNN memiliki nilai akurasi sebesar 32.9%, pada nilai  $k$  adalah

7. Sehingga ketika nilai  $k = 7$  ( $k$  yang paling optimal) metode PCA–MKNN meningkatkan nilai akurasi dari metode PCA–KNN pada saat nilai  $k = 7$  sebanyak 3.1%. Selanjutnya PCA–KNN meningkatkan akurasi pada  $k = 15$ , namun belum bisa melebihi akurasi tertinggi PCA–MKNN sebelumnya pada saat  $k = 7$  dan  $k = 15$ . Pada Gambar 4.66 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 3 fitur baru.



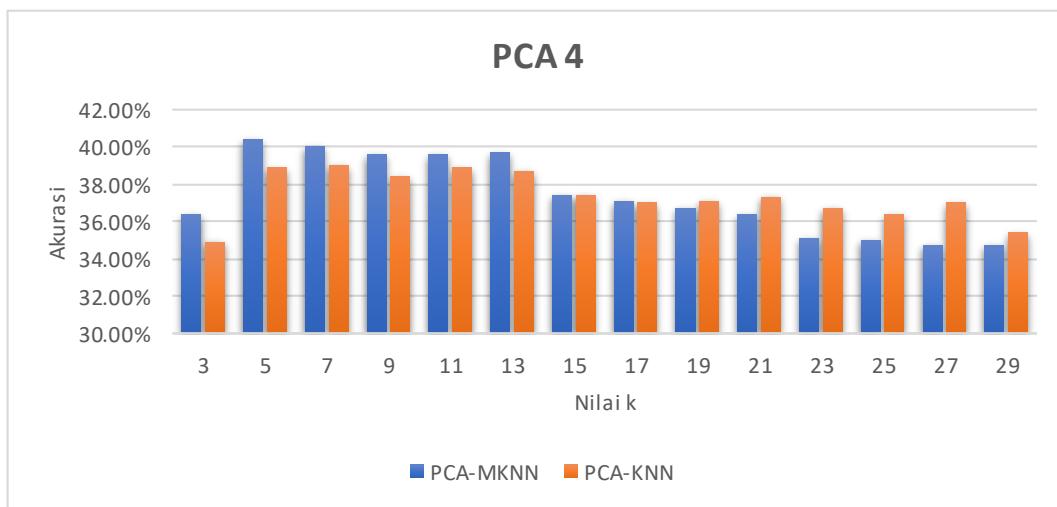
Gambar 4.66. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 3 fitur baru

#### 4.5.2.4. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 4 fitur baru

Dari keseluruhan nilai  $k$  dengan 4 fitur baru, PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan PCA–KNN. Pada Gambar 4.67 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata–rata akurasi metode PCA–MKNN meningkatkan akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 5$  dengan nilai akurasi 40.4% merupakan akurasi tertinggi pada metode PCA–MKNN, sedangkan pada metode PCA–KNN

memiliki nilai akurasi sebesar 38.9%, pada nilai  $k$  adalah 5. Sehingga ketika nilai  $k = 5$  ( $k$  yang paling optimal) metode PCA–MKNN meningkatkan nilai akurasi dari metode PCA–KNN pada saat nilai  $k = 5$  sebanyak 1.5%. Selanjutnya PCA–KNN memiliki akurasi lebih tinggi dari PCA–MKNN pada  $k = 19$  sampai dengan  $k = 29$ , namun belum bisa melebihi akurasi tertinggi PCA–MKNN sebelumnya pada saat  $k = 5$ . Pada Gambar 4.67 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 4 fitur baru.



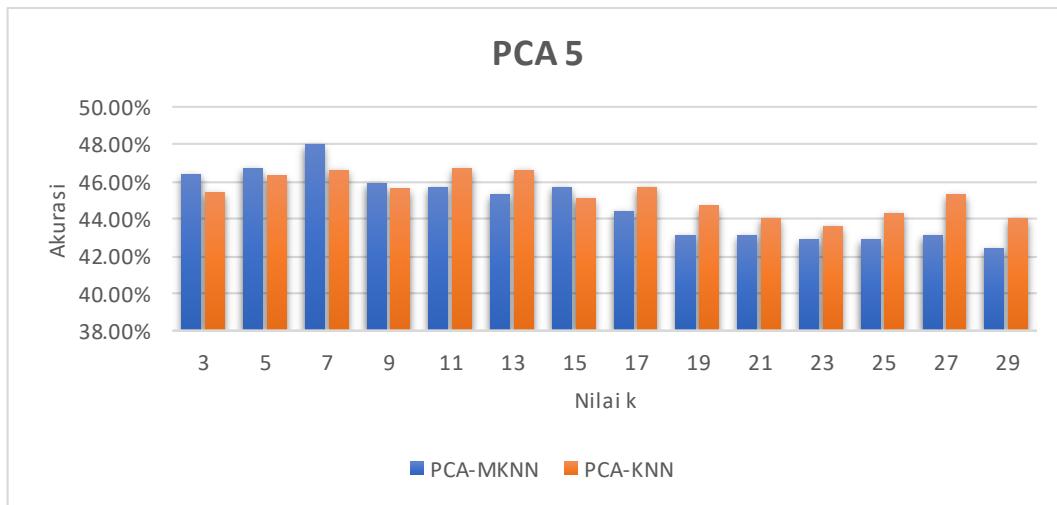
Gambar 4.67. Perbandingan Akurasi perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 4 fitur baru

#### 4.5.2.5. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 5 fitur baru

Dari keseluruhan nilai  $k$  dengan 5 fitur baru, PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan PCA–KNN. Pada Gambar 4.68 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata–rata akurasi metode PCA–MKNN meningkatkan akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih tinggi dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 7$  dengan nilai akurasi 48% merupakan

akurasi tertinggi pada metode PCA–MKNN, sedangkan pada metode PCA–KNN memiliki nilai akurasi sebesar 46.6%, pada nilai  $k$  adalah 7. Sehingga ketika nilai  $k = 7$  ( $k$  yang paling optimal) metode PCA–MKNN meningkatkan nilai akurasi dari metode PCA–KNN pada saat nilai  $k = 7$  sebanyak 1.4%. Selanjutnya PCA–KNN memiliki akurasi lebih tinggi dari PCA–MKNN pada  $k = 11$ , namun belum bisa melebihi akurasi tertinggi PCA–MKNN sebelumnya pada saat  $k = 5$ . Pada Gambar 4.68 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 5 fitur baru.

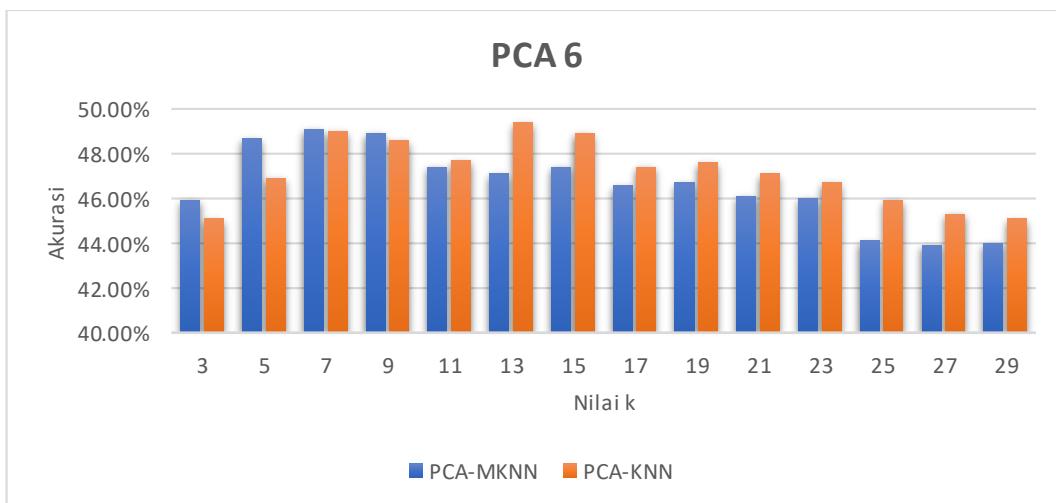


Gambar 4.68. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 5 fitur baru

#### 4.5.2.6. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 6 fitur baru

Dari keseluruhan nilai  $k$  dengan 6 fitur baru, PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA–KNN. Pada Gambar 4.69 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata–rata akurasi metode PCA–MKNN tidak melebihi akurasi tertinggi dari PCA–KNN.

Pada metode PCA–KNN memiliki akurasi yang lebih tinggi dibandingkan dengan metode PCA–MKNN. Pada nilai  $k = 7$  dengan nilai akurasi 49.1% merupakan akurasi tertinggi pada metode PCA–MKNN, sedangkan pada metode PCA–KNN memiliki nilai akurasi sebesar 49%, pada nilai  $k$  adalah 7. Sehingga ketika nilai  $k = 7$  metode PCA–MKNN meningkatkan nilai akurasi dari metode PCA–KNN pada saat nilai  $k = 7$  sebanyak 0.1%. Kemudian metode PCA–KNN meningkatkan akurasi pada nilai  $k = 13$  dengan nilai akurasi 49.4% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN mengalami penurunan akurasi menjadi 47.1% pada nilai  $k$  adalah 13. Sehingga ketika nilai  $k = 13$  ( $k$  yang paling optimal) metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 13$  sebanyak 2.3%. Pada Gambar 4.69 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 6 fitur baru.



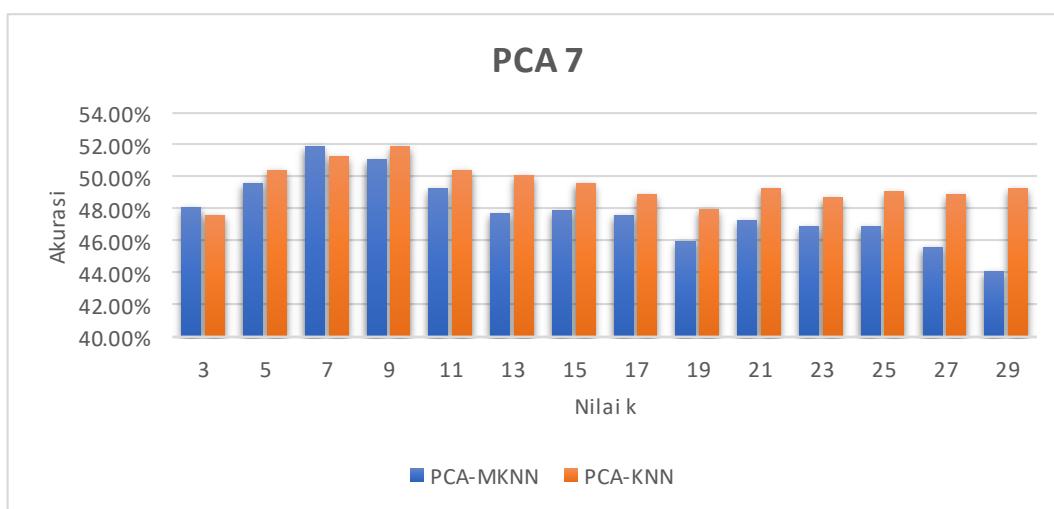
Gambar 4.69. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 6 fitur baru

#### 4.5.2.7. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 7 fitur baru

Dari keseluruhan nilai  $k$  dengan 7 fitur baru, PCA–MKNN memiliki akurasi tertinggi yang sama dengan PCA–KNN. Pada Gambar 4.70 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi

mengalami tren naik, turun dan naik lagi, sedangkan hasil nilai rata-rata akurasi pada metode PCA-MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik lagi. Namun nilai rata-rata akurasi metode PCA-MKNN memiliki akurasi tertinggi yang sama dengan PCA-KNN.

Pada metode PCA-MKNN memiliki akurasi tertinggi yang sama dengan PCA-KNN. Pada nilai  $k = 7$  dengan nilai akurasi 51.9% merupakan akurasi tertinggi pada metode PCA-MKNN, sedangkan pada metode PCA-KNN memiliki nilai akurasi sebesar 51.3%, pada nilai  $k$  adalah 7. Sehingga ketika nilai  $k = 7$  ( $k$  yang paling optimal) metode PCA-MKNN meningkatkan nilai akurasi dari metode PCA-KNN pada saat nilai  $k = 7$  sebanyak 0.6%. Kemudian metode PCA-KNN meningkatkan akurasi pada nilai  $k = 9$  dengan nilai akurasi 51.9% merupakan akurasi tertinggi pada metode PCA-KNN, sedangkan pada metode PCA-MKNN mengalami penurunan akurasi menjadi 51.1% pada nilai  $k$  adalah 9. Sehingga ketika nilai  $k = 9$  ( $k$  yang paling optimal) metode PCA-KNN meningkatkan nilai akurasi dari metode PCA-MKNN pada saat nilai  $k = 9$  sebanyak 0.8%. Pada Gambar 4.70 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA-MKNN dengan PCA-KNN pada 7 fitur baru.

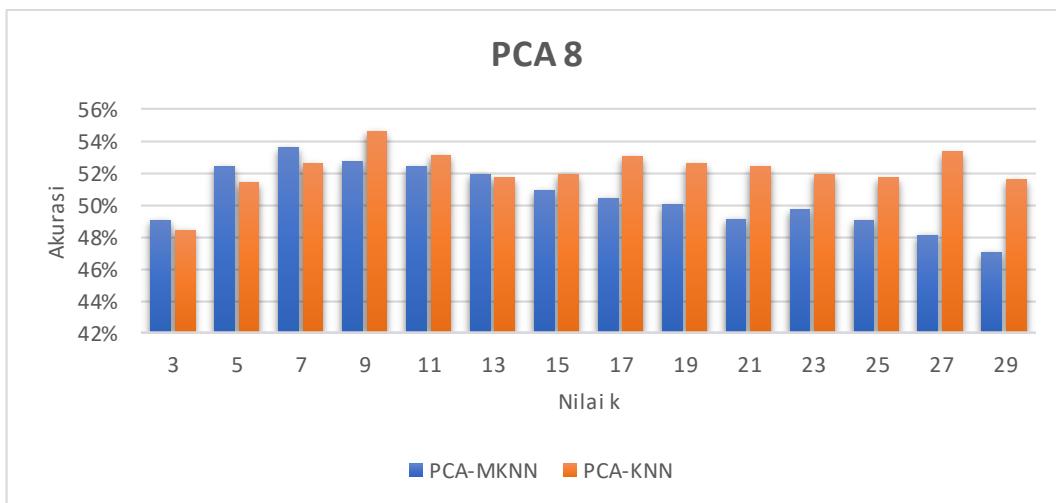


Gambar 4.70. Perbandingan Akurasi pada perubahan nilai  $k$  PCA-MKNN dengan PCA-KNN pada 7 fitur baru

#### 4.5.2.8. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 8 fitur baru

Dari keseluruhan nilai  $k$  dengan 8 fitur baru, PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA–KNN. Pada Gambar 4.71 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA–MKNN tidak melebihi akurasi tertinggi dari PCA–KNN.

Pada metode PCA–KNN memiliki akurasi yang lebih tinggi dibandingkan dengan metode PCA–MKNN. Pada nilai  $k = 7$  dengan nilai akurasi 54% merupakan akurasi tertinggi pada metode PCA–MKNN, sedangkan pada metode PCA–KNN memiliki nilai akurasi sebesar 52.6%, pada nilai  $k$  adalah 7. Sehingga ketika nilai  $k = 7$  metode PCA–MKNN meningkatkan nilai akurasi dari metode PCA–KNN pada saat nilai  $k = 7$  sebanyak 1.4%. Kemudian metode PCA–KNN meningkatkan akurasi pada nilai  $k = 9$  dengan nilai akurasi 54.6% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN mengalami penurunan akurasi menjadi 53% pada nilai  $k$  adalah 9. Sehingga ketika nilai  $k = 9$  ( $k$  yang paling optimal) metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 9$  sebanyak 1.6%. Pada Gambar 4.71 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 8 fitur baru.

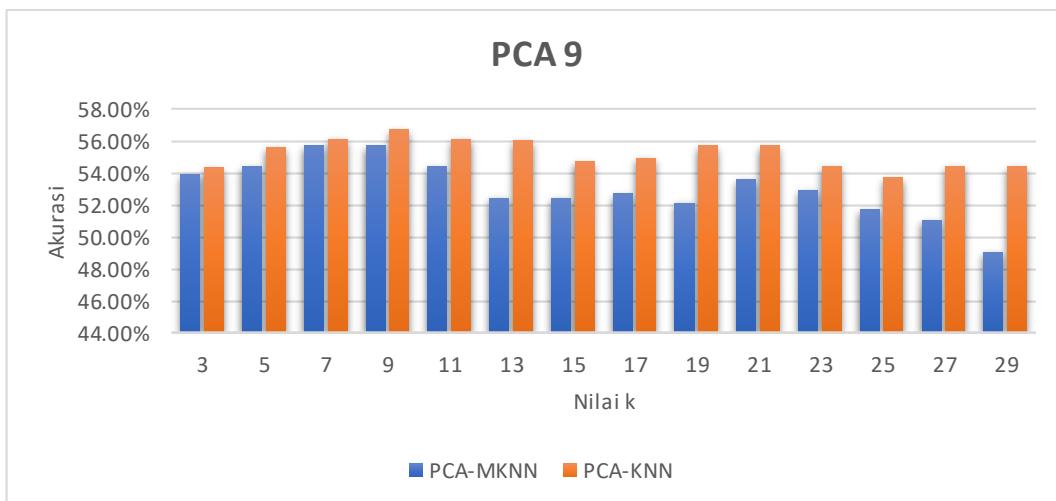


Gambar 4.71. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 8 fitur baru

#### 4.5.2.9. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 9 fitur baru

Dari keseluruhan nilai  $k$  dengan 9 fitur baru, PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA–KNN. Pada Gambar 4.72 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA–MKNN tidak melebihi akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 9$  dengan nilai akurasi 56.7% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN memiliki nilai akurasi tertinggi sebesar 55.7%, pada nilai  $k$  adalah 7 dan 9. Sehingga ketika nilai  $k = 9$  ( $k$  yang paling optimal) metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 9$  sebanyak 1.0%. Pada Gambar 4.72 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 9 fitur baru.

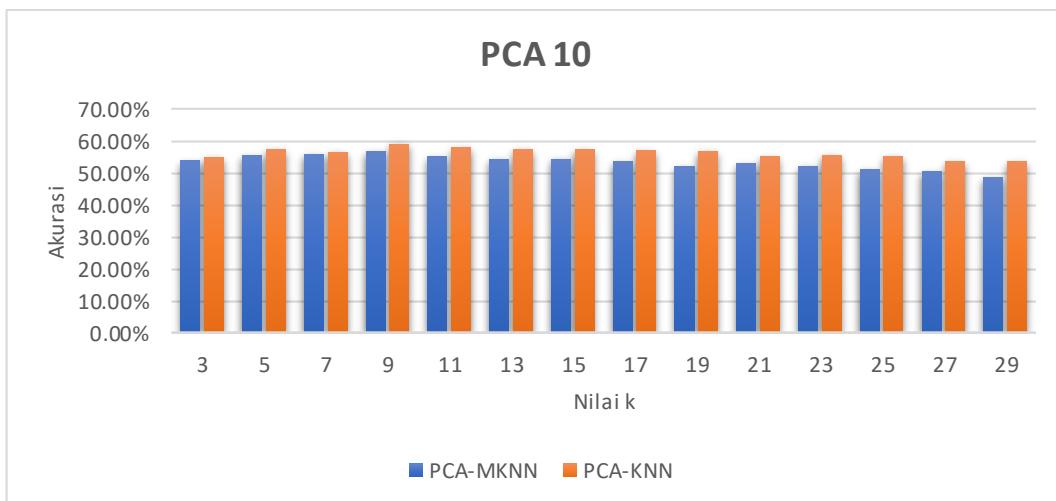


Gambar 4.72. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 9 fitur baru

#### 4.5.2.10. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 10 fitur baru

Dari keseluruhan nilai  $k$  dengan 10 fitur baru, PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA–KNN. Pada Gambar 4.73 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA–MKNN tidak melebihi akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 9$  dengan nilai akurasi 58.9% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN memiliki nilai akurasi tertinggi sebesar 56.6%, pada nilai  $k$  adalah 9. Sehingga ketika nilai  $k = 9$  ( $k$  yang paling optimal) metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 9$  sebanyak 2.3%. Pada Gambar 4.73 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 10 fitur baru.

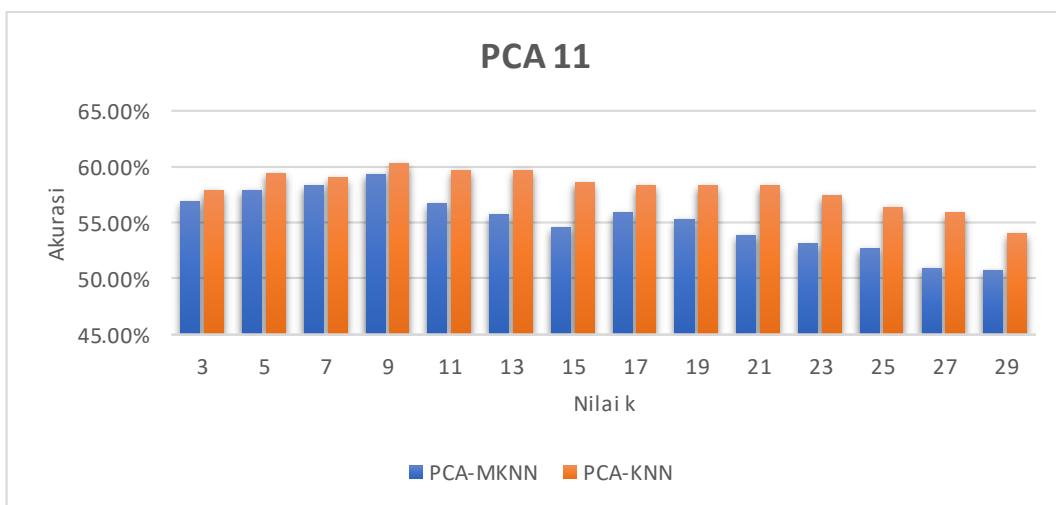


Gambar 4.73. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 10 fitur baru

#### 4.5.2.11. Perbandingan Akurasi pada perubahan nilai $k$ PCA–MKNN dengan PCA–KNN pada jumlah 11 fitur baru

Dari keseluruhan nilai  $k$  dengan 11 fitur baru, PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA–KNN. Pada Gambar 4.74 menunjukkan hasil akurasi saat  $k$  bernilai bilangan ganjil 3 sampai dengan 29 nilai rata–rata akurasi pada metode PCA–KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil nilai rata–rata akurasi pada metode PCA–MKNN cenderung juga mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA–MKNN tidak melebihi akurasi tertinggi dari PCA–KNN.

Pada metode PCA–MKNN memiliki akurasi yang lebih rendah dibandingkan dengan metode PCA–KNN. Pada nilai  $k = 9$  dengan nilai akurasi 60.3% merupakan akurasi tertinggi pada metode PCA–KNN, sedangkan pada metode PCA–MKNN memiliki nilai akurasi tertinggi sebesar 59.3%, pada nilai  $k$  adalah 9. Sehingga ketika nilai  $k = 9$  ( $k$  yang paling optimal) metode PCA–KNN meningkatkan nilai akurasi dari metode PCA–MKNN pada saat nilai  $k = 9$  sebanyak 1.0%. Pada Gambar 4.74 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 11 fitur baru.



Gambar 4.74. Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN pada 11 fitur baru

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Dari penelitian yang telah dilakukan, terdapat 3 kesimpulan yang dapat diambil. Kesimpulan tersebut yaitu:

1. 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  $9$ .
2. 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–KNN yaitu  $60.3\%$  dengan nilai  $k = 9$ , sedangkan akurasi dari PCA–MKNN saat nilai  $k = 9$  yaitu  $59.3\%$ . Selanjutnya akurasi PCA terendah pada PCA 1 fitur akurasi terendah PCA–MKNN yaitu  $17.10\%$  dengan nilai  $k = 3$ , sedangkan pada PCA–KNN saat nilai  $k = 3$  yaitu  $18.7\%$ .
3. 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  $1.1\%$ . Perbandingan  $k$  yang paling optimal dan PCA yang paling optimal antara KNN dan MKNN terdapat selisih akurasi sebesar  $1.0\%$ . 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.

## **5.2. 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 yaitu forward selection dan backward Backward selection (Putra, 2020), decision tree dan discrete wavelet transform (DWT) (Han, 2011).

## 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.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining Concepts and Techniques*. USA: Morgan Kaufmann Publishers is an imprint of Elsevier.
- Hassanat, A.B., Abbadi, M.A., & Alhasanat, A.A. (2004). Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach . (*IJCSIS*) *International Journal of Computer Science and Information Security*, 33-39.
- Hazliansyah. (2012, 09 19). *Apa Sih Musik Soul Itu?* Retrieved from REPUBLIKA.co.id: <https://www.republika.co.id/berita/senggang/java-soulnation/12/09/19/makwua-apa-sih-musik-soul-itu>. Diakses pada 24 januari 2021
- 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>
- Jamalus. (1998). *Panduan Pengajaran buku Pengajaran musik melalui pengalaman musik*. Jakarta: Proyek Pengembangan Lembaga Pendidikan.
- Jolliffe, I. (2002). *Principal Component Analysis, Second Edition*. New York: Springer-Verlag.
- Mutrofin, S., Izzah, A., Kurniawardhani, A. & Masrur, M. (2014). Optimasi Teknik Klasifikasi Modified K-Nearest Neighbor Menggunakan Algoritma Genetika. *JURNAL GAMMA*, 130-134.

- 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.
- Prasetya, E. (2012). *Data Mining-Konsep dan Aplikasi menggunakan MATLAB*. Yogyakarta: Andi.
- 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>
- Sitinjak, A.J. (2019). *PENGARUH MUSIK INSTRUMENTAL TERHADAP PRESTASI BELAJAR SISWA SMK N 1 SEI RAMPAH*. Medan: UNIVERSITAS MEDAN AREA.
- 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/>

Winarso, Bambang. (2016, April 6). *Tips & Trik Berkenalan dengan Layanan Streaming Musik, Spotify*. Retrieved from dailysocial.id: <https://dailysocial.id/post/apa-itu-spotify>. Diakses pada 24 Januari 2021

## **LAMPIRAN**

**LAMPIRAN A**  
**ID Data Yang Digunakan**

**Lampiran A.1. ID Lagu *Blues* yang digunakan.**

No	ID	No	ID
1	7b5StK8dFlp7ljUHgT95mJ	26	6HA9QpH9xzSXH6qw1FtbzD
2	1NgjW8K5Qbn7tYQW52hGgo	27	0CcySM3v6xGPdqkYFPZWZ
3	56UscliEGCYrmCTSI3veuvC	28	5tRNgEl5hyFr7E2A dt7dhT
4	6IWxHgVbdKyUMydhVGXRaT	29	6bsM19KBQ7rhFXY0CN9M8o
5	31BuUoRtbJwnpzXb4EomGR	30	1SfdzhXyrw84Hs0oMLaeEC
6	4dumXvXMddYwmRElxOjxJC	31	1uPb7QAH2vycKIWdGp6ZlY
7	0K0lbZb4fVSZUHfV4Uss9	32	2jXrwWE32uQJBi50xCvE9c
8	3v3IWe1q0LwjMR0HIN9V6b	33	3cg0dJfrQB66Qf2YthPb6G
9	7BvIryGDGfHGhZBXqDrfrY	34	2Qm98Rw4rdfvo14ZOUiZTN
10	0mBSy1Bm4I6sOhe8bjCibK	35	6Gyk7ZHfFWo3d8U7poUEPs
11	0gg5d9CX3nwZV7Hx2Mnuj	36	7cOzgDiU8MLB8IkK1RsWn1
12	0bobD7WpzTRZesNV8C8CER	37	1Msf1vU7lpnbqUDxj6HpnW
13	4zBSdZ9E6FWnj7Ks64xdNZ	38	5zGCD8uXutwg pKmXYA VYDb
14	1lCQALskBTmdt1Hlsqc3oA	39	3OHBGcvkS6oYtCVlxLXP6h
15	4QGwPf8fEs72KTSXnPfPHl	40	46tOHo9SThlggrU2MiijEC
16	4behsNa8x0dwysWBn6gUHS	41	1Q57HueU3dO9hZB5ALMIPh
17	2klvIQb1m0faNZ83PoFB7x	42	2O7QjNCjqflzWOTVg9UIEm
18	4FlmEEneXpRCQ0EbFjOYQ5	43	5cDIbJtehjhImG8eytKwGw
19	1i7nQGgwloILcp6ZftH2Qa	44	0utSSFLHA1lfTpfnS7PtS
20	3gv5or6v2B1cJMFB1rHJN	45	4xu0LW34dFEoHG1x65tQPx
21	5j6tQBjNNG4wLLdVISJ5g2	46	1OwuyjjWet6BpgIHSEfnNS
22	7ATNFqcr0pTZuFendjxtpk	47	6ihFHZeKzXHlrbY2thI4Vx
23	7tXVTjc7OpVMGPXXkxK7GI	48	401zrDasfDlsAPE7ADGfkO
24	2glwN5raCeEn0EdjKrNw6i	49	1cyn40tHPYeyXaLQ5mVhlb
25	7v6GgmyNIsNwLBPsp1pzdG	50	5YU67a01EqyrZGlxqmVuCV

**Lampiran A.2. ID Lagu *Classical* yang digunakan.**

No	ID	No	ID
1	7qNlmpT3ANKIOTiT9J79Vx	26	1DtoFS423q4L3whcGRXJjX
2	70mZwr5zctUTtltrA5T2N4	27	2Yjf0o7ytpQ5EcotNJK2ID
3	2oLjhx7w8Hyd3gry9cCXr7	28	3GiSy33Qr6cHArDIJ3Hgrh
4	5IM7q5auY0dzqD8jxnujl	29	7Ci53Nn5otFGaccKNRm5qP
5	4jep6dMPP0z7HrN375InoL	30	2NPuqw0MFpDe5vccgQesT2
6	23tA7m6wooo2XHJ8YacENk	31	0FwoESTITuc9leMP6KtYSg
7	18tYhf6O5pUMtMnYC8iA6B	32	3a1FmTzjPWRFlh3BE54ff3
8	0S6KWNO0BRK2qs1iclhSHr	33	086sjLP EqdKBgTx bTeCLCv
9	0LmBOXeyjnX1zie6AIEQmy	34	1L4NadMj3Jj7YgGaikyhfr

10	7AXul8TGnkgrWaMXUNevTq	35	4JYFwr8TFFEiA5Ao5Q7ZS7
11	0l68eSURjXY7oO2eKttQ5	36	1i8tzGQ59LjKqs1DzmAs4z
12	0iFgzKBaKbRbG5XJ9ZXLC	37	3oHSL6pt9LpNrQZuQGu9wL
13	3kBtqDJvBkxnmu xo5cQlkj	38	5X4QnmLUFC7AWs xG4tteAm
14	0AQJWtbXycfuzKdXLrBuim	39	2SgQu9bIq8O6EBd3Bm8N8L
15	22saDmEaFYv0sZsVkJZMRNg	40	76ODIwEMw8cuSif210qpb8
16	3h5LXb9kR0qPlpMCO1xfDw	41	0ftAOThcYwpSDapJhDOy7o
17	41sr0e0L4r15xTTbU2K0Gx	42	5nioVBwH3xs v10UWdVts 10
18	5vvNdNx1i8mL32zsqnvtgd	43	4zEGECYEuEPAq52wHs wpPB
19	0uUH9tgwwflMXT3XCiQ8NI	44	0uL3g8pq0ZXexhQC0Q4pFA
20	6S43QzI1j4znIKuefdpraQ	45	1kxhcIVisIav4uxDoDL8gB
21	76CE8P9vX7ev3zjVQmLprs	46	5ejSIiseOusMj76vBVnB5Y
22	2e8MxBgVWMSQmxb2zcuCoq	47	6GzMz3s0K1YKwRVI36CgRx
23	0uwAWC1AQCXfKEoUnC1mcO	48	2xizRhme7pYeITbH1NLLGt
24	1t46GkU2H4x6zG41DrDk7	49	294lzNQIRp4s oeVaQhows7
25	1ipJdiNMjgvfAilddGT9sI	50	7h6GoPvGHC9uzZJ8bNvfIq

### Lampiran A.3. ID Lagu Country yang digunakan.

No	ID	No	ID
1	0Nu9WA8kEbBW Esay2s8Q0U	26	0ttKO7eBjyoq5Khr2X9lwV
2	0gv4JVcgmb8HihoucEXVxh	27	6dGWq08LXuIw6T2oUeHECh
3	1OmKo4t4Bh95xQI6WGiUR3	28	4SBeexFIGHjl o8oK6jUQcw
4	4c68Z9wLdHc36y3CNjwQKM	29	0VwTeYNjcl30DyQlt3GPe0
5	3M1aZdmyIqNNFPE8hOKZTZ	30	0MHRECaT0Hzq4iqJutJLtz
6	2y849Gnu4VoCnUNaxCh51T	31	7ledv9CTjqdwnvHgRmETyZ
7	3rXCZRMiMZp0feGcYXpwYX	32	4TnUKixNW MfajncgdSwFoi
8	6sySDQamLYi4wSONik95Rv	33	5Tqb1bof64uhOIRRbqnZkF
9	0DUrdV917olRPxTesJAI3n	34	57fGFSzxpGVosCuWWJeKTv
10	5cXdB92xagJr4b30GjRbfX	35	5dNGORzx Fb5z3v7vSAFkTU
11	6ANPIv5r3vdWntrmFa6H6M	36	3OjNkFFZavF89xvRqWCXmU
12	1ZGu1039OcTLbDDsFKuY0j	37	63u5QGp9dKXAoYTVKUKBYn
13	6JHQBkPuKXCM6j4l2tPqc1	38	6BCrbWBpb8d6KWmEqZ41tr
14	0mG66FLhO0iZ6XCA Gdj pax	39	4f8hBeMXMvssn6HtFAtblo
15	6xatfNMI8NkY5XxRHAeiS4	40	4afTRi1HGNqIPum5Y9QeOk
16	73XG0kGl tI3jaHlg xoC5u1	41	6vmAg12y9Mp oZKrVUXrPe5
17	1OtiKIQ3EALplQW w2LF7Xn	42	4yEvfGgJ9tYfyfXXMLza1V
18	7gNXxA vRBZxBhf v0HSUF2z	43	3SRuODYNV4j2ckkJ9Samju
19	7qbXjgIPXSQW4mNYP3Yk93	44	2dptTvC3yH12TGtzqp79qw
20	1A6P8Ix zNyBRsQualaNXNY	45	5ExWY6ojo1Qm3rC7iaEqx2
21	0JSkrv29sBBohFK3qZTEQ0	46	2MiXC2LfpsoeUP5RvMGEQ3
22	6VXNavmpQoWMoUgI6hyLrc	47	1KNNTdw7SzJ90p6RXq4kGE
23	1AEHl2gbtSrdKoLjZX2Y7J	48	71WaAqNtNKNFcHZ2scb9Bi

24	5Sqw0uCMrc2Xam8hXfMMj	49	3wKfQaOgFxZ0qhXjo4iqll
25	3oyByfBYegVEDwYyq9pRd	50	2jZ0RcOfjdhoqvylgqNrvh

#### Lampiran A.4. ID Lagu EDM yang digunakan.

No	ID	No	ID
1	3DogAzf7yKYC4gQ5JESchV	26	5knrAEjUMh6T2kDCN7YD2b
2	6X21dCXcpbkKCXtKt8EsqN	27	1vvNmPOiUuyCbgWmtc6yfm
3	6pSKM75L1PjoFol5KUWiRC	28	6nDKrPlXdpomGBgAlO7UdP
4	1rCwIWmn9qTb9VTFQNqKWm	29	525GY0mXYnxajolwgS8Eaz
5	1wjM0BcLE9oSoT8q5UT81y	30	0OJN2A3Qyvd7pwSF0AIteC
6	5fVllp6Ph4IZwlKAjugfa5	31	5kgqTe1BM720OjU78TGyDw
7	6hT4NNMHBsQLCIJLMwMGGW	32	6HUnnBwYZqcED1eQztxMBN
8	2OddRE3xHqnryjEqpa3jmJ	33	1CUVN2kn7mW5FjkqXTR2W1
9	5mt94BsIrLXRYh1NP87M2D	34	1Cq23W4ZxHTY8QbP40qjEc
10	4q4HD6OY9aYXEfpPfnfpP4	35	6JleyZldNFQxcwBL3LgF95
11	3YWf1HtkESNY0Se64USwe	36	7tUTCcqkMCnC9qLqbQel0T
12	0oYWZEjkI86WujE1GfiNZr	37	20McUFi8KCIvNdNLrkTDuZ
13	5d7RQFLEiFt1WqOUbEOweq	38	3KcUTtzg9W5Hse7RknCSZr
14	5v6aFZhEVli5GOdYG00hQb	39	6dedqDWmv9pSbK3kgF18kv
15	2i3lVJNtV5ZNkMo5dHu9AD	40	0eC9r6hu1gAcLkZpSdMi2X
16	242WIEmgCkOR1PkAmW DfPI	41	3PKfvuv5veJLbC9p4WJvDbG
17	5pA6rJ5RAgmXw9QIubTpFk	42	5BgDs2jxeM8cbypzpIGs3z
18	5ieQrVW2U70NFMg28mzlqC	43	2gnqqL572vfyt2WPHMjKhO
19	3HNs1UAn7OZQiby7DXTIYI	44	46jN5ft1b89T6x8WTq646j
20	66W1rVTnEv86dIkFhoiElg	45	2CNOujOA6SenLRhAfzoWKx
21	2J6hyfM0x5imlV7lMHwpX2	46	45ohxHHub9fgz7Oe3Fw0aG
22	1HLyQQ3nxbaXwTewU1NCJ1	47	6mCPTANMqjQBCYm04mTIUp
23	0J2vHg5G6Cl7mH5NUK6e1w	48	1LLFoZZ306GC6LZweCtJNQ
24	2qzArAFF1cIZJRqWu00ELn	49	3oD19X7tn8TvRWsqMCp6H
25	5PNvgiKSwMdjBsDsgFCFLX	50	7jvZl4szVJrNuCRowmhy8Q

#### Lampiran A.5. ID Lagu Funk yang digunakan.

No	ID	No	ID
1	17YDXy3iaj6De4pcCyxlGB	26	3KGM2fm1K3p5alOyHGzTyY
2	3xq9srnNN0LBvha15ZWpTU	27	3YTEsBzprIpelpYwv1fTs8
3	35kDqdxX71QMxGN9TfN8Ze	28	20vAA0sHoffri9wLRsnaQ2
4	5XgkwPu4UjZ6MHXp2kJgkd	29	6d4XP98nTswxMycT6fdZNN
5	4tbKutO3WSFhNyLaUs3F3s	30	0D7L4pBEGv8ni1cRfaTdMJ
6	21tzl7QqFs2UJwaGQpwtCk	31	3koCCeSaVUyrRo3N2gHrd8
7	4mvVLWbe50QZg4QJhzBztu	32	4yKZAckuudvfd600H2dQie
8	3ZQZUODE0CoMcgnlwZ7cK9	33	2M2WJ7gBlcKNxdhyfPp9zY

9	1r0nn0QF5tNJDtzoIj2mIB	34	2zMJN9JvDlvGP4jB03l1Bz
10	5qenhYD4y nkEEHDDwqBiEG	35	13v3s iPyvy5TTEZYmGPPse
11	4UdJg615OBpJu6MepS6Hq8	36	0rc9Uwm4v mnQEfx5omLWWF
12	2INsiycUgobLSy4ksO4MzK	37	6zts tiyZL6FXzh4aG46ZPD
13	3UEh2di2w2D4gfws velMn4	38	2dCmGcEOQrMQhMMS8Vj7Ca
14	6EOAOGl s9pIdUJsrqytcXK	39	5lA3pwMkBdd24StM90QrNR
15	1tFcIERJBVvo4bmtLNSC8W	40	3kim6v8wY7HLPaNm w99SxD
16	1TB0dZqFRQsnf4fLLRKQV	41	5XVb0qfm mkmms yy7j8oObk
17	4o1yx2u3jBmR360WIXZ3L0	42	4pNiE4LCVV74vfIBaUHm1b
18	7jm qTPoxOQsC2w9Cb6z78q	43	4gRA0i5sxx3jAhHaVjPnUN
19	7oYHMmBbUF1zDnbgVJwCvC	44	3mRM4NM8i07UBqrSigCQFH
20	6dokyiv4HEy6WWxQCQcOkA	45	6nJh9dyel0o2jmlZzY Gh3h
21	46eu3SBuFCXW sPT39Yg3tJ	46	28NBmfocOzTPEb6OYA9fW
22	1ELKkbWqtKRwdCBGvETBp	47	0MH Xrq n909p0LRTPsNsGEi
23	0KDc8l4VNTu6OkMYK9D7HC	48	1qiQduM84A0VeH8Y2uAbqi
24	3qQVUOHJdgIFWJd0jrG9GE	49	1i2ZI3tXuJUEjCY6zF3G9w
25	3iHLh7um7wGHjBlBVJMA2R	50	7kigmgx2tJJ sZH Kaa2QC0w

#### Lampiran A.6. ID Lagu *Gospel* yang digunakan.

No	ID	No	ID
1	1f0ULNih5OLKZk6wcLnoVx	26	6G7VwsjxNBPjCyPmzrUS2R
2	6NO0Up4grjlWcprt v4H4dN	27	6v6qYTGVq8zTgHX98SJ7Es
3	78jqNs62Tfo3sIxCL2IpI	28	5hOWTy31TqmFUOGXLcnYQJ
4	50rAuUPCO4xCW4cYGY6gRE	29	0FeRlc9264zC1Uo9PdZVf3
5	0GMUJWrXFia s kh8KYbnyw	30	3CC8a83A9FRZ5TUjpONqFq
6	4qL8V4pTMJX0HMfyTFWuin	31	05vp57A03E3V6arWWy34yL
7	3BLj3BUTTINjVkh dR45z9e	32	6v9gJ3FeLvhfuInXzSsXpc
8	4AgLJxOkacISI93JYPNUtp	33	7wkJJPm4cJnevULnXYJhJm
9	4ndpiHFCHD9vMT39QFzkXU	34	04MPnUpNYCSCllb5I2geRn
10	35CMJA PksNx7mYVmVKqmio	35	29IGrhmUFETOPkG62HCZsg
11	6tzwFtw4AHo2tCexPGMzUN	36	0ekM6xK9H98mJn4c0d3VoC
12	5bSIH6IVzXjxnQII sTPRDn	37	2LgIbzBj1WCt2Ik5tb1w4
13	3V8cEPvDkFmQh6nUJnG318	38	4srwFa8t4eIoIMEdGPVFwu
14	4wc8FpW9XCUaoxtii3CiCf	39	6v wfHkpyngJeJJxAkA6EoS
15	2BiCH1atqXX1OXOtL0PKHS	40	0yay0aA5OwXHYWBww4xuGr
16	7hZRgQiZEKfle8KTEVlkQN	41	4pxjZGz9Wk07s mCueT7je3
17	1jVpiofvq33NgRUg x0MpNc	42	0eDajJT pZNVXPUCqza0HZ
18	7thaxfNdPMpWZuqye9AZfX	43	3TI8Bp0MSS9ldP15FdFCqZ
19	6rZ18bk8Mipm5T5jmDBqJy	44	1qZ1YyF1uSa4hQliWlMzH3
20	7HkD5zNeGmDP0ufwr5dgrQ	45	3EBAbFPNcVNDXsgtSnSHDt
21	4Smg3uR1jwMr58fmaTTWJC	46	6KvFfXGLtVvW7ZYXJEvvjI
22	5AMAF69gvK0v9YT8V6csae	47	2DsTxMYWUOx8NW5hdCEGhm

23	3ZLuMo6PrHfOxmD1dLr9c7	48	6BgWo34t4HQm4cXpNgeTeF
24	2RwzuFDm2uqTaMMXXSZeSw6	49	0c4UVGcDiYxFMH0ly0xa4R
25	3UnV6ebZwFHLD276t24jbG	50	2lWFN5HviKVkFtybv85FaU

#### Lampiran A.7. ID Lagu Hip Hop yang digunakan.

No	ID	No	ID
1	78gUO21zFcruzuccs9yEOxL	26	3XNlk20lYum0okywiW3ry
2	5YEoZOojehCqxCQCCQiyR4	27	3R9j8urSPiBbapNbyuSYkE
3	0VgkVdmE4glld66l8iyGjgx	28	49WcLDJOqgp13yYkRM8AHL
4	19gEmPjfqSZT0uIDRfjl0m	29	4l59QWcA9F9qqvNc5pwDtL
5	07sOl7WsmCaUqaJaRn9Dss	30	2l8nEO0vsTRCRnGqQty6qx
6	4qikXeISRKvoCqFcHLB2H2	31	2kuZwHNMI6KkMzEcZqrfm1
7	5ujh1I7NZH5agbwf7Hp8Hc	32	4IO8X9W69dIQe0EC5ALXhq
8	7t5KciWtIWIS0MXLGqGA VZ	33	5D7LFnpYhd5jlvEWIYLkBS
9	5GATP9D34th0D88vIKi38F	34	5yY9lUy8nbvjM1Uyo1Uqoc
10	3J87xd1Wkf5YYFXXtwOnQO	35	7kRVWQ1G4o8w3ubfNW2ZHB
11	3NQhVW YQOF2kHetWDRi1ol	36	3WyRgi8CzQnhzO0xw79tTS
12	40pX8i9q4xjOWZR4rGmk2q	37	1jQvUmf16VE9WQCPuK2e8K
13	79s5XnCN4TJKTVMSmOx8Ep	38	4TgHt7vKCimpwyaiKfl0uj
14	5Rgxp3xVqijmMhByj4urIm	39	1qDLTUxg7xyylgtynleuruw
15	44ZJ4hLD5onikKs VLQaILr	40	4Y8uVzBLydQgZUPUxxakGJ
16	119c93MHjrDLJTApCVGpvx	41	69CQe4ekUZ7txcSfEXYFW9
17	1pyeIeIBIJ8VAJpF0pNPu0	42	1Si3sqUkvIj6Cz9JYO14OS
18	5RGXoZt1qxmA ErdhbyKJKq	43	4adrh1YMaPAhtqalhG8hIH
19	4IYKjN1DrYzxKxt0umJqsG	44	15Zwn2QjNDVI7xX62tuGVb
20	00wjJH0T9jU2l108mHW5Hn	45	3TFO78Jhs0NbQUrFkfGdIo
21	57hCSu4zM TdnSum7NBL1Ye	46	4ledi94TlaB2GGb1nMB68v
22	6wSwl9xDx28F6bYi7y4JBR	47	3JKgcAa7r07ocVWcV8bS0H
23	3WcGdhN6wvGplwVyixNgfI	48	5FVsbdWfHWILVy0if3IRJ4
24	4iiWcajF1fEUUpwcUewc464	49	0bKz9WAXw6Ct1cHWb2zdFc
25	5CcHZZeJAeNW64vnbYGFyM	50	0AVTLkvG3QiVSmlcTW1pVd

#### Lampiran A.8. ID Lagu Jazz yang digunakan.

No	ID	No	ID
1	2ZitZzpm4i01kZm1nNPRPg	26	6V5OcaQior9q7a8r4b93M5
2	0QBgBYJGKbQ94uBKSts zv0	27	3yFlBWuW488dH4Ar8IoV7T
3	1e5Xhxz3Ij8TLCNuhB2IEL	28	3oVvVcIpGSDKv2klHewYsC
4	2m6G2vP3x33rQhfmSYB9Iy	29	2o82FM3VkoceMB2UxrYAoa
5	1JcaZ6CL3xI6f4s Yv9hNbJ	30	7sRYWbFAFNfK59is0lP5lc
6	0LZ678lgdb9f0ab8AyZBHH	31	55o3rbQcbBVXUqrVs zu5rC
7	3cAnCLivD2rxdgHN1B27vX	32	2s0lOju4ZXSGnLQeKyRehk

8	01Y1rFJpmKXucS9yNdwin0	33	5pKuAjYVNyUzIgGO6uk0H1
9	4zyETWIER8end95J1QlzXN	34	0XX58klcfvomYeXTyVUa4b
10	29nK4itJtykZEtSVE8dt4R	35	4W4rgYePwQXWaJMSVGAHBz
11	74tBxy0qDxFviVy77F4oDK	36	6huWhUGSRCegtm2fvw4Wbc
12	4xMSs10bUKFuWmOs2DmBzw	37	4iXAEGvve4JekS5RgiwEZB
13	6lgrssuSPxoY9r0RlqMPiH	38	7Kl3QKrZVfQTMnLFrMy4Q4
14	5jD6uLibX1R7wKGfJ6zx4N	39	3i112QUFH4sxGGU0iYFamK
15	0khBfVz8DAHfM3GiHAjMMk	40	0ymty2BkJI50ZYQagxpzt
16	09N0r6vIBIEbqcpuAYvKEL	41	5lPhB7GM1jyY1CcnNC2KD0
17	0K98aZItWBqCnFiMMiXiD8	42	1lvIHUFBQoESWGDqTJaxhN
18	30qPlOpLeJWD1mA SyON1po	43	2KFIDeoTAuwLrXI0ok4jvL
19	3JluMpRA2buMEgVjFU8YMj	44	60ToqY3gAOYleyxX3B13gW
20	50ma4LjaP5XDL1BsHrZ1D9	45	3RpjDd6ITzQAsqG101BFH8
21	7wHsraPGARpUvcRW8holDn	46	3FNtP8FngAuXkPEi4HaFyC
22	4Z3UHUWnGLHD0qh f6gSjCO	47	3EeJwALV0aHIW3rIamN6aM
23	4GMJNNA04YDBVixAWiK5zD	48	0eo87ddUocYp59TJL6fdks
24	1snRw3CQImnuTq7UaX6Are	49	4ENhcl2XYNbXqhiE3eBzon
25	4cUBqqGzo3cOpbUnFDcPX2	50	1Fb9kAuV11nIwb8dVlbvpF

#### Lampiran A.9. ID Lagu Metal yang digunakan.

No	ID	No	ID
1	5ITT2UsqV9R5jRnpUYebcA	26	5lAXI4GRCeBJYAX821Q329
2	2roO1GmcaLE9iBs0x2D0P0	27	6qpaNtUKnf9ykKHqHdBhxN
3	6QopguYpChUZ4Kp8v4Sm20	28	2T9hjqYQNAChUAqaa9gIRyE
4	6tJfZIv8cKoi8dGs aGuZv4	29	33LekxKniOCW zuSUT4HH6f
5	2j4XwCIwdzs0HsMnnB28Sh	30	3Uh8cFbZM YQkeyPT40K2u3
6	3pIQ0DcZjEhWTlu3BPBT9s	31	115WwncSW6yWMcgr93UWX C
7	4TFBJMB8TBX3Z3psYnptHL	32	1UmcktVwMddeYqyNfL9WXu
8	1MEBTRzovGtBfHBsOJJC3	33	2KaDrkD7r8xSgAfHls xi9U
9	0CqCpm4LLQbHUX5ysisWJS	34	7jbEwkIEcmqtAeYEw2k2fB
10	54MXF9I8s3DuiQo3g0gZ5k	35	5wq1sRE1xStyFMYibf7IA2
11	6UevPknPIGDy71vgGtZdzP	36	0Dr6tWieJeJ1irEjWteIAv
12	6oIS0TmHmsGr0hXtcBsiVM	37	0ZWbQZ3tlI7X7XNplTXGX4
13	6jNjY9mDWROqXDJmU6AXnj	38	3Hgn2dn7YMLhsJKIH1BOWa
14	4IRvl2PMPqvavdm4Q1cNV0M	39	2knnewYZIIaElrwOtXA Tqc
15	0mlUS9GApGWW8wGxNI0CPE	40	34XcuHrlxzt8pTfkZEivwn
16	2AjEKcRXkntPkujLXp6hGD	41	23WfrNnWRNPkOqFcwtrjo6
17	3wrUHyog7VOsCYWHjOnYn1	42	0IvF88MI22klaFnOjuSYmQ
18	5bOmleKvMS956pVl9uBLaI	43	1p3733Z0Nv6mg1kMHhSj4W
19	18o4pskv7PvFzOWVhShDDh	44	3P1yWZrm1FoDAsEHkBON7
20	5Ouu2mxQIsBMC51fhVAbs	45	7D5nmKDtgj6wP9nWqG4LMk
21	7sAAonuVO7tqy1q69X6ZUj	46	5Bog7FbfP8VAqZRBpoic2y

22	0iJ8HLpc5I1EifrCho1hPQ	47	0TGtojDTWbq2cTNXJ8R2jM
23	178VAE5qT0LSfs wN5rTNPu	48	2ok3k2i4xhAbTq45fQ4r65
24	2GKvYk3zl0011ZhbuhdnYv	49	4lsTgoftlh7NR240tfAV82
25	3SqIytsA0RzewQYCJhtIFm	50	2b8OC2ibUQD6ILvEJ6AmBd

#### Lampiran A.10. ID Lagu Pop yang digunakan.

No	ID	No	ID
1	4w1lzcaoZ1IC2K5TwjalRP	26	4cbJwuAEbaodP4InQDfAmW
2	7tbJozWeewmFvTkXCUFtt0	27	3TGRqZ0a2l1LRblBkJoaDx
3	3S0OXQeoh0w6AY8WQVckRW	28	3XVBdLihbNbxBwZosxcGuJ
4	6lanRgr6wXibZr8KgzXxBl	29	0InJPuWIYp8fwlBGIQglLf
5	1MDoll6jK4rrk2BcFRP5i7	30	3cNjgVBKTJ1SvKhunrCdVy
6	3tCwjWLicbjMCvXhN0WOE	31	5LqanqJSmZwbwPtIi2RwTb
7	3AFfyUXTfj12PgrOkZiiJo	32	6BoyG3bQ8rjDdqatMg5IW M
8	3E7dfMvvCLUddWis suqMwr	33	6cR3LxWz9Nx57EpZCaNGb4
9	0mUyMawtxj1CJ76kn9gIZK	34	2AEepQ8zdGMSaSYrm3pq0W
10	4E6cwWJWZw2zWf7VFbH7wf	35	3w0YTGr6S3ntMDnvxC8c6N
11	4PFAm88FBkY7IFtvCaGctZ	36	4msighkG4nK0w6aoN7IV7
12	1lkvpmrCaXK8QtliFDcHBO	37	24wAuTYk4J4WDzVRyaEBH2
13	4lLtanYk6tkMvooU0tWzG8	38	6l5ckcpGw0izUlJFwkgk21
14	4fIWvT19w9PR0VVBuPYpWA	39	2BA7LobrOd4nofvdTmrHI7
15	43wROOsAEK0F3Fu46Vjn7W	40	5A69ahUutZ8FWEOoXswkAw
16	6ECp64rv50XVz93WvxXMGF	41	7er7WofjVw4QwPAyxH39jH
17	4FCb4CUbFCMNRkI6lYc1zI	42	6cG3UfwxKQPZlh7qMN3Lhe
18	754kgU5rWscRTfvIsuEwFp	43	0OjiV7FpVCrOlq53ZSH1EM
19	33LC84JgLvK2KuW43MfaNq	44	28pSjEpLxPmXMr0EzqqqTP
20	6OuN3su7IVAD7YvHZiAt8l	45	0fiKmieZ6xzoolzNOC3Fxi
21	2uyspZLecy9ZSoz27c6JTm	46	2Yg72UW2YSwIdb4WFeuupn
22	09bzKNKOKC2YfJZsGMW GtF	47	71IfOcA83ywKh3tbDFbqcw
23	7KwnrVNUEAkIZFqEc0FM9Zg	48	7vZTvo2aJ6wHKCGjwjdWIf
24	0rgEL2cD2T5MDzSDJTQNlw	49	0JHmZ5JKMkm06BQIjdZ4At
25	3TLmBfWx4IC6FJsZGLV1v8	50	1BaGfrllWinaF77p7QASBH

#### Lampiran A.11. ID Lagu R&B yang digunakan.

No	ID	No	ID
1	7N28LOrx4qiEkL1Zd3OTZ3	26	50DAHcCqneoupFyh96P2GP
2	5YLnf7R2kueN0BRPkjiEG	27	0ZTJrZ1axCUMP1fiFzyabF
3	35w6fQ0Wnuv7aIryRx7gTq	28	0cfn5OBGafI32fEsc3z4GE
4	2eBD7tEXYvIxrxANueGwtg	29	69mFxANpGyq9s69LqH3d9G
5	6JnMvs ohvCY0s htvTQCSHd	30	1knipGsDFp6FBFc6aRRyjL
6	2LHRwhsMByGIVTtHxlOJg2	31	6Vas6h9lB5mmeVDXdFqlQA

7	1RDLbX9L2G4zDqxXCaMJo5	32	4jnFqNWeJCeCRHc4HCdxfd
8	1FKFLWFa7fVoftc7UmYhDC	33	4LxIGAVfcQIw0zAQRyFhU8
9	1i1ofxObE2LhvDk6mmm3Fo	34	6mz1fBdKATx6qP4oP1165G
10	2K9Ovn1o2bTGqbsABGC6m3	35	51RvCRzeTqoQgdOBgcpHfT
11	0KS2h61pHQ4WmOwruD7uxD	36	4j1Bk0BEIGCF9hR7cSwl9d
12	1LV5cAo02H8h5YZNcJULM	37	5IFCyWplye09HytlP80RCF
13	4OmCPaP0sOh2FxuAFdLYzJ	38	7tsRp7QKUBp6hc9bth0h7x
14	5XqDJFVCyRTm5J7cIfRmR1	39	6xkryXuiZU360Lngd4s x13
15	71hBQ9qHsZU2M3vDF8MtaN	40	0JCAezTC5IIK1a9o0YBEsD
16	08HPWLi0YmRUStJzkQ1ffh	41	17aVK3oQdgwVG2l0HpP67W
17	27PNPVQqaqQ0m2p7uoxnFo	42	5A0aUvnfAEcmUx3hoXjI8R
18	7KzLjZBZwm8ikKqs wryXWK	43	22RCzyBU4gHnZ6pdJzT3RM
19	4cOVTA2GfYTHw99AJDQpHo	44	2eHj0klWkwRQuIrNIPpCPa
20	0ri0Han4IRJhzvERHOZTMr	45	6g0Ors xv6gITJCt4cHsRsQ
21	3mxXvmaDJXzjZb9WMZVo3L	46	1sZLO7n7ap5F6w9KhCRPPm
22	3XG801WWhqC753dekEBkMt	47	2lp8xjq0WTm3HZKHuDeweg
23	6QBw7YZlbpo4C9kRIGns NL	48	7fs4v7plEu5CIs oOt9lLk8
24	1aWV3uY3SIEZVbmv45oFWS	49	4eoli5b55cvAkvwP4tB1TF
25	47TqCCnEliDp8NRDyIQoQq	50	5Od8o3WsbeLxlGP1vmH6s6

**Lampiran A.12. ID Lagu Reggae yang digunakan.**

No	ID	No	ID
1	4pbQqdxHDI1oJpKxEfHxvq	26	260F1LrhRMiuXOZIaypMZd
2	07QuKUTu5O40AibYYPx98I	27	1KgsVjds vVf68nEm8lzvFO
3	7HXcSZsNnjm5HV49Rlk7n4	28	6vAdiI54nLUVHXkw2uMMTS
4	3Gym3Rtm6FHpTrLJdTcz3j	29	79yFFWjMYYNFq3MHosWlly
5	1JSiX4OT6JuJATNIUA0hL2	30	4SGRc3MuFpYmFQ9Z8wal9F
6	7uJGYKzr2dRoxU10AJlSbP	31	2SE11vmjbi7j5QKJ885ib6
7	6mILiMvNURTw98oGpMRYJP	32	7jut3RkkxQpJAzMovUefDz
8	6j6DxAHlmrqPLb21BLi7zM	33	34pB4FSvt545dwYbpbRkgd
9	7BfW1eoDh27W69nxsmRicb	34	7z3EXUkDIgi6y3L46JSklf
10	1haHZa75ytryLGUOtHiGl5	35	1AO26gK1sI6PoFY9eUF8yV
11	6D4eMFcg3DquhNOQsNanQk	36	1YpH91YcUZlo2mxKJVhWM8
12	21QASdwUnDJ3kDuxZojWoJ	37	4MGa3uO14yw16f8IPKnLZ6
13	2PyDqCtFcXsucZqTF4rE3i	38	4fxF8ljwryMZX5c9EKrLFE
14	0I23NUlxIwV5OjdZvNkEMS	39	01yLsK4Zab603sXbIuTQ5B
15	4Qup2zholsphLpgkO77C2	40	5nDPGDOqmAeSV4hXQsAU01
16	06YsLfaNGjGk0l7Z77H1s9	41	4FNm4ZJaUVLDCMXcqwS6zY
17	0GGPUysb8GpCG76dQH63te	42	30IDv8NTGHirorcjkk8MLR
18	7gZAFQROjd8LkSBtAjPiYr	43	5O4erNIJ74PIF6kGo11ZrC
19	7BnSsZU3WPrZYj6ZjoX0iE	44	6JRLFiX9NJSoRRKxowlBYr
20	3i66eIhRTTULVB6dLJbPCL	45	4tvir8RdYs zAMkrai yTSmN

21	5lGhbmfAqIbCKIhn5QcIPg	46	49CdYBpfABUa0ZfT8FizQZ
22	0di7hMT6o1ogeUN1XDB9Ft	47	5Uv1YNz6uBepPLDdoFANKQ
23	044WMdxZWPLXL3hQJmx2We	48	0zJSkE5Nw3i3in3BBSuG
24	6Q5DtPsZY5I9LEZnjL1PDn	49	5NYWAIsnlOKNUlmKoDMENy
25	7578oiL9J6tFHXEjfCXFAS	50	0QkWubamAKEvvOTriSKX3

#### Lampiran A.13. ID Lagu Rock yang digunakan.

No	ID	No	ID
1	5anCkDvJ17aznvK5TED5uo	26	6CDhfV6K4ypIWYFR8Ip82v
2	3sHH7lkfbpcraDDvYnjo7	27	1KCAEG6JcsTzi8ddZzs0nk
3	6FVYwnVrnAEIRnY3bHJb46	28	297d7PwfSFHQXCjqqDXQT2
4	0DkmhHO4yyqCJFjosmmWPU	29	0L7zm6afBEtrNKo6C6Gj08
5	7zAt4tdL44D3Vuzs vM0N8n	30	5dTHzHFPyi8TiTtzoz1J9
6	4VymMhILk4DSwghPkLUPVS	31	7eRG7KNDIgjIk3nvymkhOT
7	58KPPL1AdLHMvR2O2PZejr	32	1f3yAtsJtY87CTmM8RLnx
8	4wH29TM74O3788Kn1KzWuY	33	4Zc7TCHzuNwL0AFBlyLdyr
9	5pKCDm2fw4k6D6C5Rk646C	34	24NXDobrJlwX9ayx7n9OnH
10	2zYzyRzz6pRmhPzyfMEC8s	35	003vvx7Niy0yvhvHt4a68B
11	4UEo1b0wWrHMC8bVqPiH8	36	5moTxUGPZXgGmosl4rIELm
12	1fQaoh3imrMunWVZh5kf90	37	4GJxfICdyvs1q0ciTILBQJ
13	2NImo2MajNai4Clk35wMut	38	7JhXQXjf79vuQoOCtFLDu6
14	5V3mdRI2yQxbSsJGDPC5ID	39	5XnyWvKPVgJsVKmUjFbMv3
15	2SgbR6ttzoNICRGQOKjrop	40	2dm3UxRHnKk2aN0oZSwULc
16	3NW1h8l6EY5N6HNBL83S4V	41	3DM2o4hRRQxdL6WMRpHoPp
17	3cwDSDzTiWr5H5xMQhQ6Mx	42	2E7W1X4maFFcjHrVrFA7Vs
18	3H0ZY55seypgoozXI6zwRk	43	5XcZRgJv3zMhTqCyESjQrf
19	5hQfs9spSyXUavvPe7jmgn	44	3QbRod3FsEsKApzRaCgiCt
20	3jagGO7eHHuaD53ibehkux	45	6nRCTb5b0N5zp8WTeY6xFZ
21	5siM4eG36axvG3allNnsAx	46	4VqPOruhp5EdPBeR92t6lQ
22	6zKF4293k44ItKWJJgrhXv	47	3KhF2YiNpJvGpfICW45R6D
23	15zUfY9L1p7bN9JsgaTAP0	48	6MtHW9npByiapSCJpE1rQS
24	3xktQXIr1OD3ENc19viwDP	49	79rFv44RxuMb70y6sHC14l
25	6RJdYpFQwLyNfDc5FbjkgV	50	01PIEMEsHPNmGEzUGYXo7y

#### Lampiran A.14. ID Lagu Soul yang digunakan.

No	ID	No	ID
1	62GYoGsQfROZswLee6W3O	26	66vDBjvjz2YvSF9yJ9JnC5
2	0OKXjd9ZpZioaHyowpQ7as	27	1RG0HipdMCcaW9xJyKiUb
3	6jtiVP3cE6NW VVgEFOkLn7	28	3SdTko2uVsxFblQjpScoHy
4	6EwrL4VXLw9IV1sG8mIK1A	29	64jxwV2kklovj7ldPsPsXF
5	1SBTxJLfZA KvDaz32tUNjn	30	1TEDAg7S7iClCBtpvaQQGA

6	6T5UJIY6dG5pYJDHqSfZvG	31	4F2QVxqYaLlsZQx70Wngct
7	3ClOMebBxeNTYCofTH24YJ	32	2j24h8ZvzouJRg2HzHvRH4
8	3M783guFs iqyc4m85YqACf	33	2dGPDI NT2l79ibSNyThzyI
9	2N5zMZX7YeL1tico8oQxa9	34	2ADfheRRkK5P2f1t8s i3Lj
10	0tZkVZ9DeAa0MNK2gY5NtV	35	2OSfEYKh lSsLx6vn4O75RK
11	39qYD4J4BKvZM Qgx fXl5bv	36	6pJE6B9aAjjAhnS6a5JzfF
12	2sXp9Qmvc7mRaDBjBgcGGi	37	0NeJjNlprGfZpeX2LQuN6c
13	4qiBDIMzW2vTK8dJKw7xD3	38	4bEtg70s rLrCqQp3Nm2T5R
14	1Lc6DaGjgnLxSGLUt2HXzC	39	7ugO7ru8CtIGmX0RIUrQZm
15	17ZX2lirfQGxbPLcn25sf7	40	4591VqUIXysNlmI5NcAIUd
16	6t34N78TlbnIWvnIwtS2me	41	44UWAmppRu j5KNHWbAASuF
17	5rc28C9Ym7zIAypx3QGcnv	42	6IxGVlmZlYIp0RUwKnR6oc
18	57jyCG2tQoLpvFMJRIPW LQ	43	2g2GkH3vZHk4IWzBjgQ6nY
19	1f08YT5uGGQPYEdJLRz4A8	44	4eGHplaq1ME8oetnTuFFF
20	3uIE49oWm621AkJVXflKqk	45	4NL8D4RQsC6ux6el3m3Lg5
21	1E7qCptwRPfvZrlDyDDgjh	46	0BCScVwlR88wcHo5KUwK47
22	6rn7yTWf7iuljTIO047WT4	47	28T4qj3m5R0qbOYA JzxK6q
23	5dXED6MP1v1qghkanirb1	48	3QEktXcsLUPfE33cRZ4y2K
24	3uwIoa2emQILz6BmH6MtiW	49	7LHfhusSfulgzuiMg7e3Cz
25	1hES47PrbU8GPTI5A9lsr0	50	6UBQONiLH8kNmmB8UrV2Ef

## LAMPIRAN B

### Pengujian Pertama

#### Lampiran B.1. *Source Code* KNN Pengujian Pertama

No	Source Code
1	<pre> public function ViewKlasifikasiKNN(\$nilaiK) {     \$hasilShuffle['nilaiK'] = \$nilaiK;     \$hasilShuffle['data'] = \$this-&gt;Data-&gt;GetAllShuffle();     for (\$f=0; \$f&lt;10; \$f++) {         \$data0 = array();         \$data1 = array();         \$data2 = array();         \$data3 = array();         \$data4 = array();         \$data5 = array();         \$data6 = array();         \$data7 = array();         \$data8 = array();         \$data9 = array();         \$data10 = array();         \$data11 = array();     } } </pre>

No	<i>Source Code</i>
17	\$data12 = array();
18	\$data13 = array();
19	\$allDataTesting = array();
20	//
21	\$dataIndex0 = 0;
22	\$dataIndex1 = 0;
23	\$dataIndex2 = 0;
24	\$dataIndex3 = 0;
25	\$dataIndex4 = 0;
26	\$dataIndex5 = 0;
27	\$dataIndex6 = 0;
28	\$dataIndex7 = 0;
29	\$dataIndex8 = 0;
30	\$dataIndex9 = 0;
31	\$dataIndex10 = 0;
32	\$dataIndex11 = 0;
33	\$dataIndex12 = 0;
34	\$dataIndex13 = 0;
35	//
36	\$dataHasil0 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
37	\$dataHasil1 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
38	\$dataHasil2 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
39	\$dataHasil3 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
40	\$dataHasil4 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
41	\$dataHasil5 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
42	\$dataHasil6 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
43	\$dataHasil7 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
44	\$dataHasil8 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
45	\$dataHasil9 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
46	\$dataHasil10 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
47	\$dataHasil11 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
48	\$dataHasil12 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
49	\$dataHasil13 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);
50	for(\$j=0;\$j<count(\$hasilShuffle['data']);\$j++) {
51	if(\$hasilShuffle['data'][\$j]->playlist == 0) {
52	if(count(\$data0)<5&& \$dataIndex0 == \$f*5) {
53	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
54	array_push(\$data0, \$hasilShuffle['data'][\$j]) };
55	} else{

No	Source Code
56	\$dataIndex0++;
57	}
58	}
59	if(\$hasilShuffle['data'][\$j]->playlist == 1){
60	if(count(\$data1)<5&& \$dataIndex1 == \$f*5){
61	array_push(\$allDataTesting, \$hasilShuffle['da
62	ta'][\$j]);
63	array_push(\$data1, \$hasilShuffle['data'][\$j])
64	;
65	}else{
66	\$dataIndex1++;
67	}
68	if(\$hasilShuffle['data'][\$j]->playlist == 2){
69	if(count(\$data2)<5&& \$dataIndex2 == \$f*5){
70	array_push(\$allDataTesting, \$hasilShuffle['da
71	ta'][\$j]);
72	array_push(\$data2, \$hasilShuffle['data'][\$j])
73	;
74	}else{
75	\$dataIndex2++;
76	}
77	if(\$hasilShuffle['data'][\$j]->playlist == 3){
78	if(count(\$data3)<5&& \$dataIndex3 == \$f*5){
79	array_push(\$allDataTesting, \$hasilShuffle['da
80	ta'][\$j]);
81	array_push(\$data3, \$hasilShuffle['data'][\$j])
82	;
83	}else{
84	\$dataIndex3++;
85	}
86	if(\$hasilShuffle['data'][\$j]->playlist == 4){
87	if(count(\$data4)<5&& \$dataIndex4 == \$f*5){
88	array_push(\$allDataTesting, \$hasilShuffle['da
89	ta'][\$j]);
90	array_push(\$data4, \$hasilShuffle['data'][\$j])
91	;
92	}else{
	\$dataIndex4++;
	}
	}
	if(\$hasilShuffle['data'][\$j]->playlist == 5){
	if(count(\$data5)<5&& \$dataIndex5 == \$f*5){

No	Source Code
93	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
85	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
86	array_push(\$data4, \$hasilShuffle['data'][\$j]);
87	} else{
88	\$dataIndex4++;
89	}
90	}
91	if(\$hasilShuffle['data'][\$j]->playlist == 5) {
92	if(count(\$data5)<5&& \$dataIndex5 == \$f*5) {
93	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
94	array_push(\$data5, \$hasilShuffle['data'][\$j]);
95	} else{
96	\$dataIndex5++;
97	}
98	}
99	if(\$hasilShuffle['data'][\$j]->playlist == 6) {
100	if(count(\$data6)<5&& \$dataIndex6 == \$f*5) {
101	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
102	array_push(\$data6, \$hasilShuffle['data'][\$j]);
103	} else{
104	\$dataIndex6++;
105	}
106	}
107	if(\$hasilShuffle['data'][\$j]->playlist == 7) {
108	if(count(\$data7)<5&& \$dataIndex7 == \$f*5) {
109	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
110	array_push(\$data7, \$hasilShuffle['data'][\$j]);
111	} else{
112	\$dataIndex7++;
113	}
114	}
115	if(\$hasilShuffle['data'][\$j]->playlist == 8) {
116	if(count(\$data8)<5&& \$dataIndex8 == \$f*5) {
117	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
118	array_push(\$data8, \$hasilShuffle['data'][\$j]);
119	} else{

No	Source Code
120	\$dataIndex8++;
121	}
122	}
123	if(\$hasilShuffle['data'][\$j]->playlist == 9) {
124	if(count(\$data9)<5&& \$dataIndex9 == \$f*5) {
125	array_push(\$allDataTesting, \$hasilShuffle['
126	data'][\$j]);
127	array_push(\$data9, \$hasilShuffle['data'][\$j]) ;
128	} else {
129	\$dataIndex9++;
130	}
131	}
132	if(\$hasilShuffle['data'][\$j]->playlist == 10) {
133	if(count(\$data10)<5&& \$dataIndex10 == \$f*5) {
134	array_push(\$allDataTesting, \$hasilShuffle['
135	data'][\$j]);
136	array_push(\$data10, \$hasilShuffle['data'][\$j]);
137	} else {
138	\$dataIndex10++;
139	}
140	if(\$hasilShuffle['data'][\$j]->playlist == 11) {
141	if(count(\$data11)<5&& \$dataIndex11 == \$f*5) {
142	array_push(\$allDataTesting, \$hasilShuffle['
143	data'][\$j]);
144	array_push(\$data11, \$hasilShuffle['data'][\$j]);
145	} else {
146	\$dataIndex11++;
147	}
148	if(\$hasilShuffle['data'][\$j]->playlist == 12) {
149	if(count(\$data12)<5&& \$dataIndex12 == \$f*5) {
150	array_push(\$allDataTesting, \$hasilShuffle['
151	data'][\$j]);
152	array_push(\$data12, \$hasilShuffle['data'][\$j]);
153	} else {
154	\$dataIndex12++;
155	}
156	if(\$hasilShuffle['data'][\$j]->playlist == 13) {
	if(count(\$data13)<5&& \$dataIndex13 == \$f*5) {

No	Source Code
157	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
158	array_push(\$data13, \$hasilShuffle['data'][\$j]);
159	} else{
160	\$dataIndex13++;
161	}
162	}
163	}
164	\$dataTesting = \$allDataTesting;
165	\$jumlahBenar = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
166	\$jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
167	for(\$j=0;\$j<count(\$dataTesting);\$j++) {
168	\$hasilShuffle['dataTesting'][\$f][\$j] = \$dataTesting[\$j];
169	\$hasilDistance = array();
170	for (\$k=0;\$k<count(\$hasilShuffle['data']);\$k++) {
171	\$sama = 0;
172	for(\$l=0; \$l<count(\$allDataTesting);\$l++) {
173	if(\$allDataTesting[\$l]->no == \$hasilShuffle['data'][\$k]->no){
174	\$sama = 1;
175	break;
176	}
177	}
178	if(\$sama == 0){
179	\$kalkulasiDistance = new MyDistance();
180	\$kalkulasiDistance->no = \$hasilShuffle['data'][\$k]->no;
181	\$kalkulasiDistance->distance = sqrt(
182	pow((\$dataTesting[\$j]->danceability) - (\$hasilShuffle['data'][\$k]->danceability), 2) +
183	pow((\$dataTesting[\$j]->energy) - (\$hasilShuffle['data'][\$k]->energy), 2) +
184	pow((\$dataTesting[\$j]->key) - (\$hasilShuffle['data'][\$k]->key), 2) +
185	pow((\$dataTesting[\$j]->loudness) - (\$hasilShuffle['data'][\$k]->loudness), 2) +
186	pow((\$dataTesting[\$j]->mode) - (\$hasilShuffle['data'][\$k]->mode), 2) +
187	pow((\$dataTesting[\$j]->speechiness) - (\$hasilShuffle['data'][\$k]->speechiness), 2) +
188	pow((\$dataTesting[\$j]->accousticness) - (\$hasilShuffle['data'][\$k]->accousticness), 2) +

No	Source Code
189	<pre>         pow(( \$dataTesting[\$j]- &gt;instrumentalness) - (\$hasilShuffle['data'][\$k]- &gt;instrumentalness), 2) +         pow(( \$dataTesting[\$j]- &gt;liveness) - (\$hasilShuffle['data'][\$k]-&gt;liveness), 2) +         pow(( \$dataTesting[\$j]- &gt;valence) - (\$hasilShuffle['data'][\$k]-&gt;valence), 2) +         pow(( \$dataTesting[\$j]- &gt;tempo) - (\$hasilShuffle['data'][\$k]-&gt;tempo), 2) +         pow(( \$dataTesting[\$j]- &gt;duration) - (\$hasilShuffle['data'][\$k]-&gt;duration), 2));         array_push(\$hasilDistance, \$kalkulasiDistance);     } } usort(\$hasilDistance, "cmp"); for(\$k=0; \$k&lt;\$nilaiK; \$k++) {     \$hasilShuffle['hasilDistance'][\$f][\$j][\$k] = \$hasilDistance[\$k]; } \$hasilClass = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0); for(\$k=0;\$k&lt;\$nilaiK;\$k++) {     \$hasilClass[\$hasilShuffle['data'][\$hasilDistance[\$k]-&gt;no-1]-&gt;playlist] +=1; } \$max = \$hasilClass[0]; \$classHigh = 0; for(\$k=1;\$k&lt;count(\$hasilClass);\$k++) {     if(\$max&lt;\$hasilClass[\$k]) {         \$max = \$hasilClass[\$k];         \$classHigh = \$k;     } } \$hasilShuffle['dataHasilTesting'][\$f][\$j] = \$classHigh; if(\$classHigh == \$hasilShuffle['data'][\$dataTesting[\$j]-&gt;no-1]-&gt;playlist){     \$jumlahBenar[\$hasilShuffle['data'][\$dataTesting[\$j]-&gt;no-1]-&gt;playlist]++; } else {     \$jumlahSalah[\$hasilShuffle['data'][\$dataTesting[\$j]-&gt;no-1]-&gt;playlist]++; } switch (\$hasilShuffle['data'][\$dataTesting[\$j]-&gt;no-1]-&gt;playlist) {     case 0:         \$dataHasil0[\$classHigh]++;         break;     case 1: </pre>

No	<i>Source Code</i>
224	\$dataHasil1[\$classHigh]++; break;
225	case 2: \$dataHasil2[\$classHigh]++; break;
226	case 3: \$dataHasil3[\$classHigh]++; break;
227	case 4: \$dataHasil4[\$classHigh]++; break;
228	case 5: \$dataHasil5[\$classHigh]++; break;
229	case 6: \$dataHasil6[\$classHigh]++; break;
230	case 7: \$dataHasil7[\$classHigh]++; break;
231	case 8: \$dataHasil8[\$classHigh]++; break;
232	case 9: \$dataHasil9[\$classHigh]++; break;
233	case 10: \$dataHasil10[\$classHigh]++; break;
234	case 11: \$dataHasil11[\$classHigh]++; break;
235	case 12: \$dataHasil12[\$classHigh]++; break;
236	case 13: \$dataHasil13[\$classHigh]++; break;
237	}
238	}
239	
240	
241	
242	
243	
244	
245	
246	
247	
248	
249	
250	
251	
252	
253	
254	
255	
256	
257	
258	
259	
260	
261	
262	
263	

No	Source Code
264	\$hasilShuffle["DataMatriks"][\$f][0] = \$dataHasil0
265	;
266	\$hasilShuffle["DataMatriks"][\$f][1] = \$dataHasil1
267	;
268	\$hasilShuffle["DataMatriks"][\$f][2] = \$dataHasil2
269	;
270	\$hasilShuffle["DataMatriks"][\$f][3] = \$dataHasil3
271	;
272	\$hasilShuffle["DataMatriks"][\$f][4] = \$dataHasil4
273	;
274	\$hasilShuffle["DataMatriks"][\$f][5] = \$dataHasil5
275	;
276	\$hasilShuffle["DataMatriks"][\$f][6] = \$dataHasil6
277	;
278	\$hasilShuffle["DataMatriks"][\$f][7] = \$dataHasil7
279	;
280	\$hasilShuffle["DataMatriks"][\$f][8] = \$dataHasil8
281	;
282	\$hasilShuffle["DataMatriks"][\$f][9] = \$dataHasil9
283	;
284	\$hasilShuffle["DataMatriks"][\$f][10] = \$dataHasil10;
285	\$hasilShuffle["DataMatriks"][\$f][11] = \$dataHasil11;
286	\$hasilShuffle["DataMatriks"][\$f][12] = \$dataHasil12;
	\$hasilShuffle["DataMatriks"][\$f][13] = \$dataHasil13;
	\$calculateBenar = 0;
	for(\$j=0; \$j<count(\$jumlahBenar); \$j++) {
	\$calculateBenar += \$jumlahBenar[\$j];
	}
	\$hasilShuffle["HasilAkurasi"][\$f] = \$calculateBenar/70*100;
	}
	\$this->load-
	>view('ViewKlasifikasiKNN', \$hasilShuffle);
	}
	}

### Lampiran B.2. Source Code MKNN Pengujian Pertama

No	Source Code
1	public function ViewKlasifikasiMKNN(\$nilaiK) {
2	\$hasilShuffle['nilaiK'] = \$nilaiK;
3	\$hasilShuffle['data'] = \$this->Data->GetAllshuffle();
4	for(\$g=0; \$g<10; \$g++) {
5	\$data0 = array();
6	\$data1 = array();
7	\$data2 = array();

No	<i>Source Code</i>
8	\$data3 = array();
9	\$data4 = array();
10	\$data5 = array();
11	\$data6 = array();
12	\$data7 = array();
13	\$data8 = array();
14	\$data9 = array();
15	\$data10 = array();
16	\$data11 = array();
17	\$data12 = array();
18	\$data13 = array();
19	\$allDataTesting = array();
20	//
21	\$dataIndex0 = 0;
22	\$dataIndex1 = 0;
23	\$dataIndex2 = 0;
24	\$dataIndex3 = 0;
25	\$dataIndex4 = 0;
26	\$dataIndex5 = 0;
27	\$dataIndex6 = 0;
28	\$dataIndex7 = 0;
29	\$dataIndex8 = 0;
30	\$dataIndex9 = 0;
31	\$dataIndex10 = 0;
32	\$dataIndex11 = 0;
33	\$dataIndex12 = 0;
34	\$dataIndex13 = 0;
35	//
36	\$hasilData0 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
37	\$hasilData1 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
38	\$hasilData2 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
39	\$hasilData3 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
40	\$hasilData4 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
41	\$hasilData5 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
42	\$hasilData6 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
43	\$hasilData7 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
44	\$hasilData8 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
45	\$hasilData9 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
46	\$hasilData10 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);
47	\$hasilData11 = array(0,0,0,0,0,0,0,0,0,0,0,0,0);

No	Source Code
48	\$hasilData12 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
49	\$hasilData13 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
50	for(\$j=0;\$j<count(\$hasilShuffle['data']);\$j++) {
51	if(\$hasilShuffle['data'][\$j]->playlist == 0){
52	if(count(\$data0)<5&& \$dataIndex0 == \$g*5){
53	array_push(\$allDataTesting, \$hasilShuffle['da
54	ta'][\$j]);
55	array_push(\$data0, \$hasilShuffle['data'][\$j])
56	};
57	\$dataIndex0++;
58	}
59	if(\$hasilShuffle['data'][\$j]->playlist == 1){
60	if(count(\$data1)<5&& \$dataIndex1 == \$g*5){
61	array_push(\$allDataTesting, \$hasilShuffle['da
62	ta'][\$j]);
63	array_push(\$data1, \$hasilShuffle['data'][\$j])
64	};
65	\$dataIndex1++;
66	}
67	if(\$hasilShuffle['data'][\$j]->playlist == 2){
68	if(count(\$data2)<5&& \$dataIndex2 == \$g*5){
69	array_push(\$allDataTesting, \$hasilShuffle['da
70	ta'][\$j]);
71	array_push(\$data2, \$hasilShuffle['data'][\$j])
72	};
73	\$dataIndex2++;
74	}
75	if(\$hasilShuffle['data'][\$j]->playlist == 3){
76	if(count(\$data3)<5&& \$dataIndex3 == \$g*5){
77	array_push(\$allDataTesting, \$hasilShuffle['da
78	ta'][\$j]);
79	array_push(\$data3, \$hasilShuffle['data'][\$j])
80	};
81	\$dataIndex3++;
82	}
83	if(\$hasilShuffle['data'][\$j]->playlist == 4){
84	if(count(\$data4)<5&& \$dataIndex4 == \$g*5){
85	array_push(\$allDataTesting, \$hasilShuffle['da
	ta'][\$j]);

No	Source Code
85	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data4, \$hasilShuffle['data'][\$j]); } } else { \$dataIndex4++; } } if(\$hasilShuffle['data'][\$j]->playlist == 5){ if(count(\$data5)<5&& \$dataIndex5 == \$g*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data5, \$hasilShuffle['data'][\$j]); } } else { \$dataIndex5++; } } if(\$hasilShuffle['data'][\$j]->playlist == 6){ if(count(\$data6)<5&& \$dataIndex6 == \$g*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data6, \$hasilShuffle['data'][\$j]); } } else { \$dataIndex6++; } } if(\$hasilShuffle['data'][\$j]->playlist == 7){ if(count(\$data7)<5&& \$dataIndex7 == \$g*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data7, \$hasilShuffle['data'][\$j]); } } else { \$dataIndex7++; } } if(\$hasilShuffle['data'][\$j]->playlist == 8){ if(count(\$data8)<5&& \$dataIndex8 == \$g*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data8, \$hasilShuffle['data'][\$j]); } } else { \$dataIndex8++; }

No	Source Code
120	\$dataIndex8++;
121	}
122	}
123	if(\$hasilShuffle['data'][\$j]->playlist == 9){
124	if(count(\$data9)<5&& \$dataIndex9 == \$g*5){
125	array_push(\$allDataTesting, \$hasilShuffle['
126	data'][\$j]);
127	array_push(\$data9, \$hasilShuffle['data'][\$j]);
128	}elseif{
129	\$dataIndex9++;
130	}
131	if(\$hasilShuffle['data'][\$j]->playlist == 10){
132	if(count(\$data10)<5&& \$dataIndex10 == \$g*5){
133	array_push(\$allDataTesting, \$hasilShuffle['
134	data'][\$j]);
135	array_push(\$data10, \$hasilShuffle['data'][\$j]);
136	}elseif{
137	\$dataIndex10++;
138	}
139	if(\$hasilShuffle['data'][\$j]->playlist == 11){
140	if(count(\$data11)<5&& \$dataIndex11 == \$g*5){
141	array_push(\$allDataTesting, \$hasilShuffle['
142	data'][\$j]);
143	array_push(\$data11, \$hasilShuffle['data'][\$j]);
144	}elseif{
145	\$dataIndex11++;
146	}
147	if(\$hasilShuffle['data'][\$j]->playlist == 12){
148	if(count(\$data12)<5&& \$dataIndex12 == \$g*5){
149	array_push(\$allDataTesting, \$hasilShuffle['
150	data'][\$j]);
151	array_push(\$data12, \$hasilShuffle['data'][\$j]);
152	}elseif{
153	\$dataIndex12++;
154	}
155	if(\$hasilShuffle['data'][\$j]->playlist == 13){
156	if(count(\$data13)<5&& \$dataIndex13 == \$g*5){

No	Source Code
157	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data13, \$hasilShuffle['data'][\$j]);
158	}
159	}else{
160	\$dataIndex13++;
161	}
162	}
163	}
164	\$dataTesting = \$allDataTesting;
165	\$jumlahBenar = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
166	\$jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
167	for(\$j=0;\$j<count(\$dataTesting);\$j++) {
168	\$hasilShuffle['dataTesting'][\$g][\$j] = \$dataTesting[\$j];
169	\$hasilDistance = array();
170	for(\$k=0;\$k<count(\$hasilShuffle['data']);\$k++) {
171	\$sama = 0;
172	for(\$l=0; \$l<count(\$allDataTesting);\$l++) {
173	if(\$allDataTesting[\$l]->no == \$hasilShuffle['data'][\$k]->no) {
174	\$sama = 1;
175	break;
176	}
177	}
178	if(\$sama == 0) {
179	\$kalkulasiDistance = new MyDistance();
180	\$kalkulasiDistance->no = \$hasilShuffle['data'][\$k]->no;
181	\$kalkulasiDistance->distance = sqrt(
182	pow(((\$dataTesting[\$j]->danceability) - (\$hasilShuffle['data'][\$k]->danceability), 2) +
183	pow(((\$dataTesting[\$j]->energy) - (\$hasilShuffle['data'][\$k]->energy), 2) +
184	pow(((\$dataTesting[\$j]->key) - (\$hasilShuffle['data'][\$k]->key), 2) +
185	pow(((\$dataTesting[\$j]->loudness) - (\$hasilShuffle['data'][\$k]->loudness), 2) +
186	pow(((\$dataTesting[\$j]->mode) - (\$hasilShuffle['data'][\$k]->mode), 2) +
187	pow(((\$dataTesting[\$j]->speechiness) - (\$hasilShuffle['data'][\$k]->speechiness), 2) +
188	pow(((\$dataTesting[\$j]->accousticness) - (\$hasilShuffle['data'][\$k]->accousticness), 2) +

No	Source Code
189	<pre>                 pow(( \$dataTesting[\$j]- &gt;instrumentalness) - (\$hasilShuffle['data'][\$k]- &gt;instrumentalness), 2) +                 pow(( \$dataTesting[\$j]- &gt;liveness) - (\$hasilShuffle['data'][\$k]-&gt;liveness), 2) +                 pow(( \$dataTesting[\$j]- &gt;valence) - (\$hasilShuffle['data'][\$k]-&gt;valence), 2) +                 pow(( \$dataTesting[\$j]- &gt;tempo) - (\$hasilShuffle['data'][\$k]-&gt;tempo), 2) +                 pow(( \$dataTesting[\$j]- &gt;duration) - (\$hasilShuffle['data'][\$k]-&gt;duration), 2));                 array_push(\$hasilDistance, \$kalkulasiDistance );             }         }     usort(\$hasilDistance, "cmp");     for(\$k=0; \$k&lt;\$nilaiK; \$k++) {         \$hasilShuffle['hasilDistance'][\$g][\$j][\$k] = \$h asilDistance[\$k];     }     \$weightResult = array();     for(\$i=0; \$i&lt;\$nilaiK; \$i++) {         \$dataTest = \$hasilShuffle['data'][\$hasilDistance[\$ i]-&gt;no-1];         \$hasilShuffle['dataTestingResult'][\$g][\$j][\$i] = \$ dataTest;         \$resultDistance = array();         for(\$k=0; \$k&lt;count(\$hasilShuffle['data']); \$k++) {             \$sama = 0;             for(\$l=0; \$l&lt;count(\$allDataTesting); \$l++) {                 if(\$allDataTesting[\$l]-&gt;no == \$hasilShuffle['data'][\$k]-&gt;no) {                     \$sama = 1;                     break;                 }             }             if(\$sama == 0 &amp;&amp; \$dataTest-&gt;no != \$hasilShuffle['data'][\$k]-&gt;no) {                 \$kalDistance = new MyDistance();                 \$kalDistance-&gt;no = \$hasilShuffle['data'][\$k]-&gt;no;                 \$kalDistance-&gt;distance = sqrt(                     pow(( \$dataTest- &gt;danceability) - (\$hasilShuffle['data'][\$k]- &gt;danceability), 2) +                     pow(( \$dataTest- &gt;energy) - (\$hasilShuffle['data'][\$k]-&gt;energy), 2) +                     pow(( \$dataTest- &gt;key) - (\$hasilShuffle['data'][\$k]-&gt;key), 2) +                     pow(( \$dataTest- &gt;loudness) - (\$hasilShuffle['data'][\$k]-&gt;loudness), 2) +                     pow(( \$dataTest- &gt;tempo) - (\$hasilShuffle['data'][\$k]-&gt;tempo), 2) +                     pow(( \$dataTesting[\$j]- &gt;duration) - (\$hasilShuffle['data'][\$k]-&gt;duration), 2));                 array_push(\$hasilDistance, \$kalDistance);             }         }     } } </pre>



No	Source Code	
254	\$hasilClassDistance[\$weightResult[\$k] ->playlist]+=\$weightResult[\$k]->weight;	
255	}	
256	\$chooseClass = 0;	
257	\$cacheMax = 0;	
258	for (\$k=0;\$k<count(\$hasilClassDistance);\$k++) {	
259	if (\$cacheMax<\$hasilClassDistance[\$k]) {	
260	\$cacheMax = \$hasilClassDistance[\$k];	
261	\$chooseClass = \$k;	
262	}	
263	}	
264	\$hasilShuffle['dataHasilTesting'][\$g][\$j] = \$chooseClass;	
265	if(\$chooseClass == \$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist){	
266	\$jumlahBenar[\$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist]++;	
267	} else {	
268	\$jumlahSalah[\$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist]++;	
269	}	
270	switch (\$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist) {	
271	case 0:	
272	\$hasilData0[\$chooseClass]++; <td></td>	
273	break;	
274	case 1:	
275	\$hasilData1[\$chooseClass]++; <td></td>	
276	break;	
277	case 2:	
278	\$hasilData2[\$chooseClass]++; <td></td>	
279	break;	
280	case 3:	
281	\$hasilData3[\$chooseClass]++; <td></td>	
282	break;	
283	case 4:	
284	\$hasilData4[\$chooseClass]++; <td></td>	
285	break;	
286	case 5:	
287	\$hasilData5[\$chooseClass]++; <td></td>	
288	break;	
289	case 6:	
290	\$hasilData6[\$chooseClass]++; <td></td>	

No	Source Code
291	break;
292	case 7:
293	\$hasilData7[\$choseClass]++; break;
294	case 8: \$hasilData8[\$choseClass]++; break;
295	case 9: \$hasilData9[\$choseClass]++; break;
296	case 10: \$hasilData10[\$choseClass]++; break;
297	case 11: \$hasilData11[\$choseClass]++; break;
298	case 12: \$hasilData12[\$choseClass]++; break;
299	case 13: \$hasilData13[\$choseClass]++; break;
300	}
301	}
302	\$hasilShuffle["DataMatriks"][\$g][0] = \$hasilData0;
303	\$hasilShuffle["DataMatriks"][\$g][1] = \$hasilData1;
304	\$hasilShuffle["DataMatriks"][\$g][2] = \$hasilData2;
305	\$hasilShuffle["DataMatriks"][\$g][3] = \$hasilData3;
306	\$hasilShuffle["DataMatriks"][\$g][4] = \$hasilData4;
307	\$hasilShuffle["DataMatriks"][\$g][5] = \$hasilData5;
308	\$hasilShuffle["DataMatriks"][\$g][6] = \$hasilData6;
309	\$hasilShuffle["DataMatriks"][\$g][7] = \$hasilData7;
310	\$hasilShuffle["DataMatriks"][\$g][8] = \$hasilData8;
311	\$hasilShuffle["DataMatriks"][\$g][9] = \$hasilData9;
312	\$hasilShuffle["DataMatriks"][\$g][10] = \$hasilData10;
313	\$hasilShuffle["DataMatriks"][\$g][11] = \$hasilData11;
314	\$hasilShuffle["DataMatriks"][\$g][12] = \$hasilData12;
315	\$hasilShuffle["DataMatriks"][\$g][13] = \$hasilData13;
316	\$calculateBenar = 0;
317	for(\$j=0; \$j<count(\$jumlahBenar); \$j++) {

No	Source Code
331	\$calculateBenar += \$jumlahBenar[\$j];
332	}
333	\$hasilShuffle["HasilAkurasi"][\$g] = \$calculateBenar/70*
334	100;
335	}
336	\$this->load->view('ViewKlasifikasiKNN', \$hasilShuffle);

## Lampiran B.3. Implementasi Sistem pada KNN dan MKNN Pengujian Pertama

### 1. Output Input Klasifikasi

Pengguna memasukkan nilai  $k$  yang nantinya akan digunakan untuk proses KNN dan MKNN. Nilai  $k$  dibatasi hanya bilangan asli (bilangan bulat positif yang bukan 0) dan nilai  $k$  tidak boleh genap, melainkan harus bilangan ganjil untuk menghindari kemungkinan dua kelas berbeda memiliki jumlah suara yang sama. Form dari input nilai  $k$  seperti pada Gambar 1.

Gambar 1. Input Nilai k

### 2. Output Klasifikasi KNN

Output dari KNN akan munjur jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 2, merupakan hasil dari klasifikasi KNN.

PROGRAM STUDI INFORMATIKA														Normalisasi	Klasifikasi	Tambah PCA	Klasifikasi dengan PCA
<b>KLASIFIKASI MUSIK BERDASARKAN GENRE PADA LAYANAN STREAMING MUSIK SPOTIFY MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR DAN MODIFIED K-NEAREST NEIGHBOR</b>																	
<b>TANPA SELEKSI FITUR PRINCIPAL COMPONENT ANALYSIS (PCA)</b>																	
Data Setelah Normalisasi																	
Show 10 ▾ entries																	
No	Genre	Playlist	Danceability	Energy	Key	Loudness	Mode	Speechiness	Acousticness	Instrumentalness	Uiveness	Valence	Tempo	Duration	Search		
1	Gospel	5	0.541	0.675	0.0909091	0.898795	1	0.066	0.1	0	0.977	0.355	0.421518	0.429194			
2	Reggae	11	0.766	0.49	0.545425	0.719925	1	0.3	0.359	0.0085	0.046	0.959	0.0819322	0.187261			
3	Hip Hop	6	0.479	0.549	1	0.789509	0	0.373	0.57	0.0299	0.127	0.976	0.761917	0.304463			
4	Edm	3	0.873	0.847	0.0509091	0.866688	1	0.103	0.00715	0.0013	0.0802	0.385	0.401213	0.416223			
5	Country	2	0.546	0.723	0.900901	0.803866	0	0.094	0.429	0.0232	0.157	0.331	0.321975	0.263891			
6	Hip Hop	6	0.917	0.516	0.727273	0.879009	1	0.419	0.149	0	0.092	0.575	0.518089	0.169919			
7	Soul	13	0.656	0.32	0.818182	0.788154	1	0.0285	0.521	0.000294	0.159	0.64	0.0949473	0.173922			
8	Pop	9	0.499	0.797	0.181818	0.890234	1	0.075	0.0433	0.0000146	0.155	0.376	0.751388	0.362783			
9	Blues	0	0.502	0.112	1	0.601117	1	0.0427	0.977	0.021	0.104	0.272	0.482047	0.208825			
10	Rock	12	0.541	0.968	0	0.938257	1	0.0398	0.000248	0.00000828	0.423	0.505	0.42074	0.383754			

Gambar 2. Output Klasifikasi KNN

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 3, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-1.



Gambar 3. Output Klasifikasi KNN K-Fold Iterasi ke-1

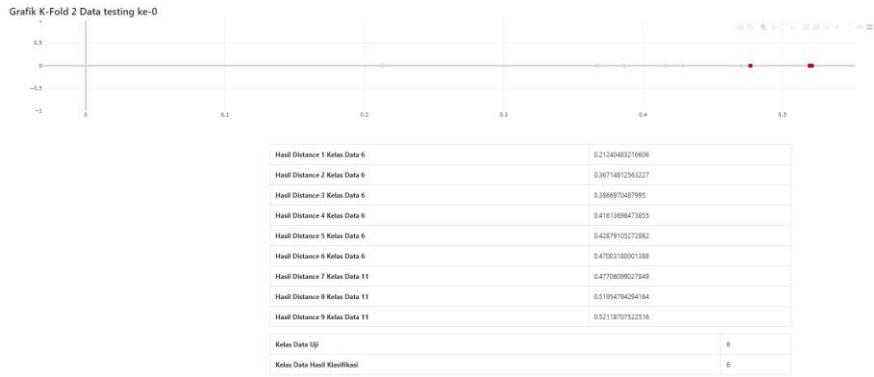
Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 4, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-2.



Gambar 4. Output Klasifikasi KNN K-Fold Iterasi ke-2  
Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 5, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-3.

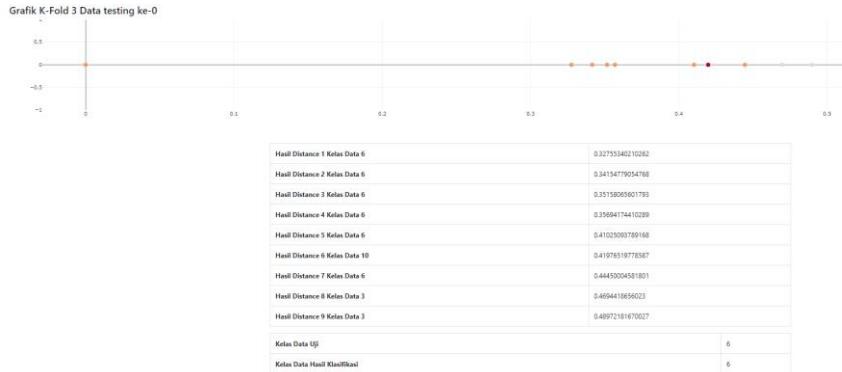
K-Fold Iterasi ke-3  
Akurasi: 61.428571428571%

	Prediksi													
\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	0	0	0	1	0
k	1	0	3	0	0	0	0	0	2	0	0	0	0	0
t	2	0	0	1	0	0	0	0	0	0	2	0	1	1
u	3	0	0	0	4	0	0	0	0	1	0	0	0	0
a	4	0	0	0	0	5	0	0	0	0	0	0	0	0
i	5	0	0	0	0	0	5	0	0	0	0	0	0	0
l	6	0	0	0	0	0	0	5	0	0	0	0	0	0
	7	0	1	0	0	0	0	0	4	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0
	9	0	0	3	0	0	0	0	0	0	2	0	0	0
	10	0	0	1	0	1	0	0	0	0	0	1	0	2
	11	0	0	1	0	0	0	0	0	0	1	1	0	1
	12	0	0	0	0	0	0	0	0	2	0	0	3	0
	13	1	0	0	0	1	0	0	0	0	0	0	0	2
TP	4	3	1	4	5	5	5	4	5	0	1	1	3	2
FP	1	1	5	0	2	0	0	2	2	3	5	0	4	2
TN	64	64	60	65	63	65	65	63	63	62	60	65	61	63
FN	1	2	4	1	0	0	0	1	0	5	4	4	2	3
Akurasi	97.142857142857%	95.714285714286%	87.142857142857%	98.571428571429%	97.142857142857%	100%	95.714285714286%	97.142857142857%	88.571428571429%	87.142857142857%	94.285714285714%	91.428571428571%	92.857142857143%	87.142857142857%



Gambar 5. Output Klasifikasi KNN K-Fold Iterasi ke-3

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 6, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-4.

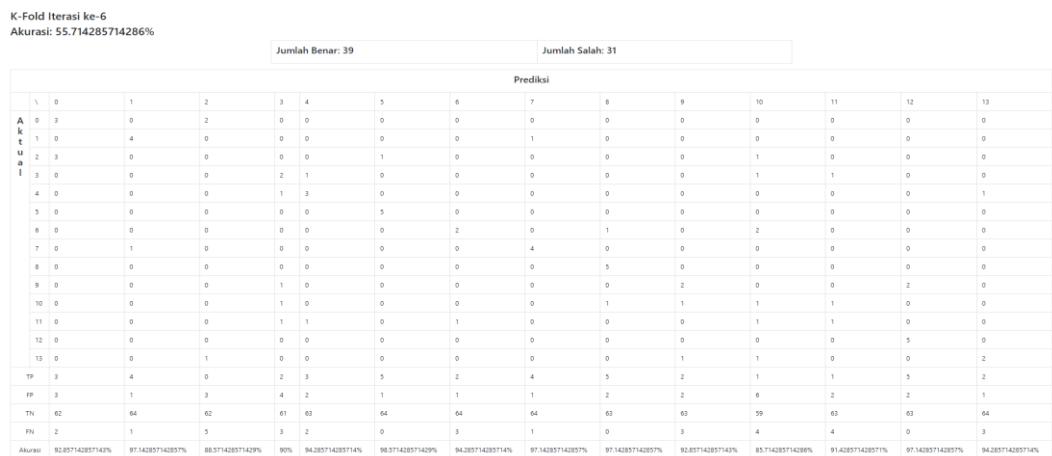


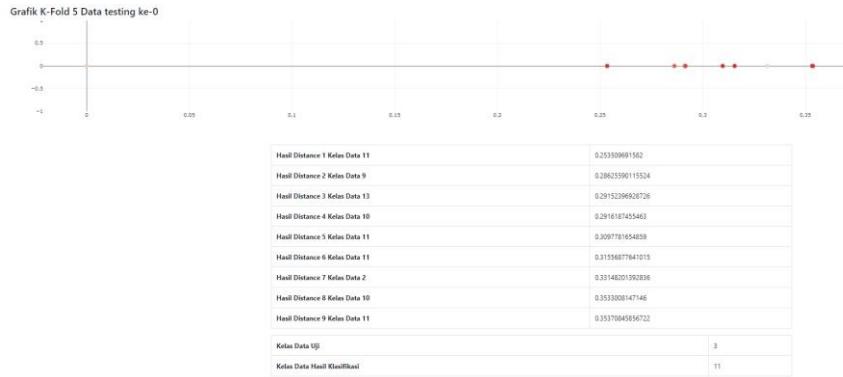
Gambar 6. Output Klasifikasi KNN K-Fold Iterasi ke-4

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 7, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-5.



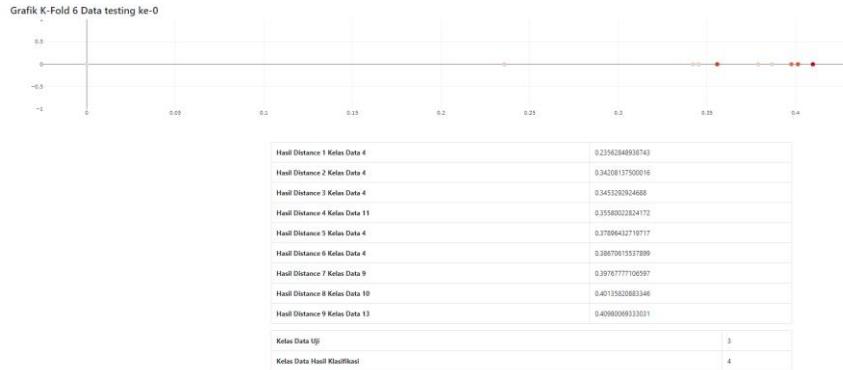
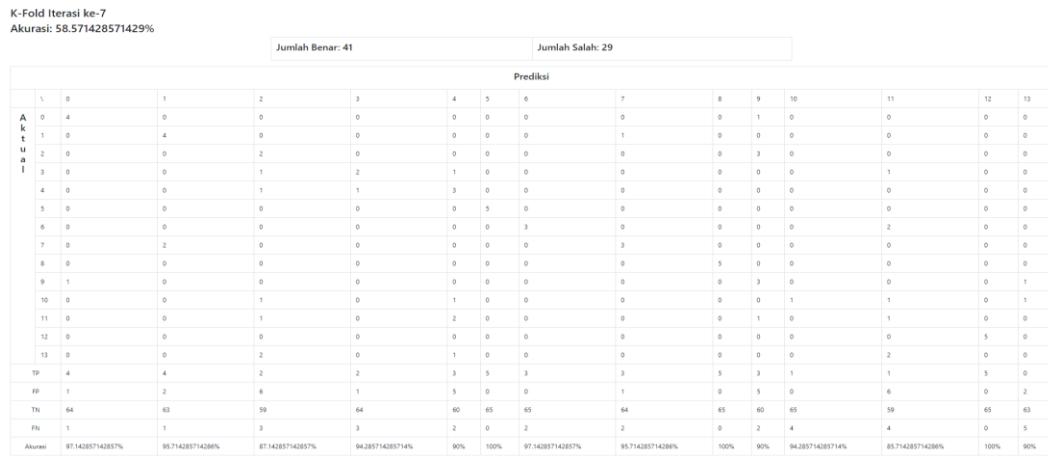
Gambar 7. Output Klasifikasi KNN K-Fold Iterasi ke-5  
Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 8, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-6.





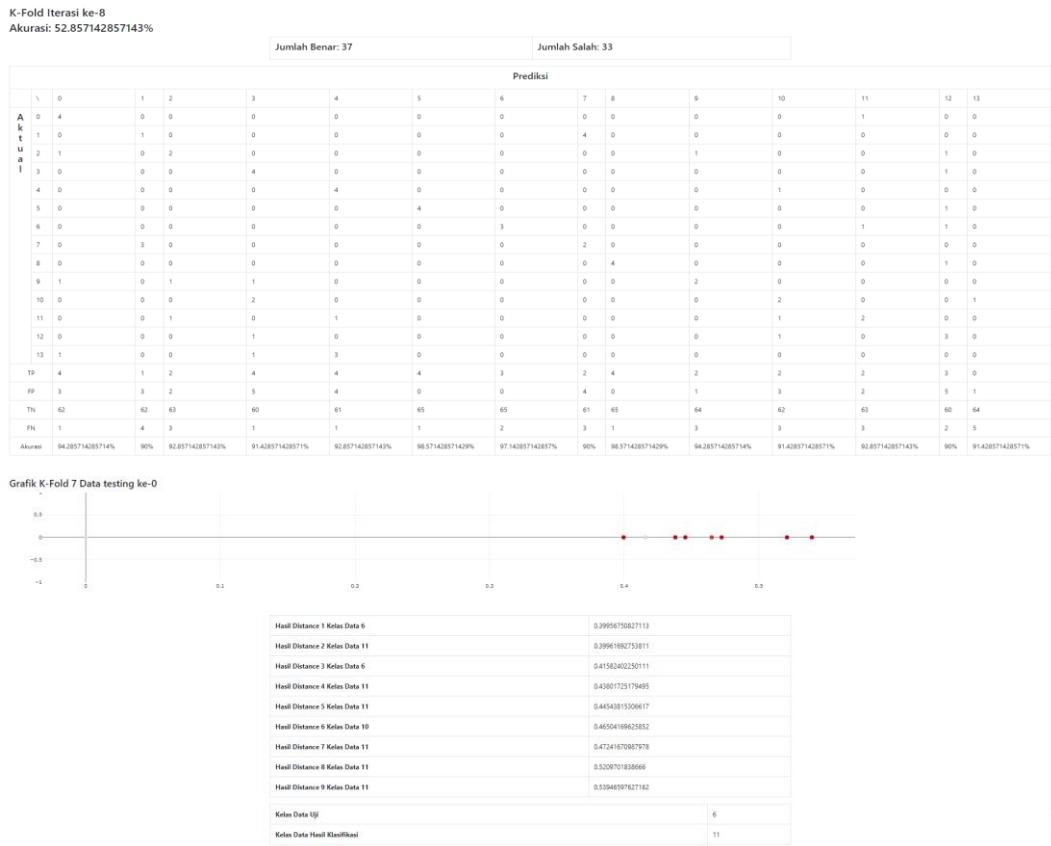
Gambar 8. Output Klasifikasi KNN K-Fold Iterasi ke-6

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 9, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-7.



Gambar 9. Output Klasifikasi KNN K-Fold Iterasi ke-7

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 10, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-8.

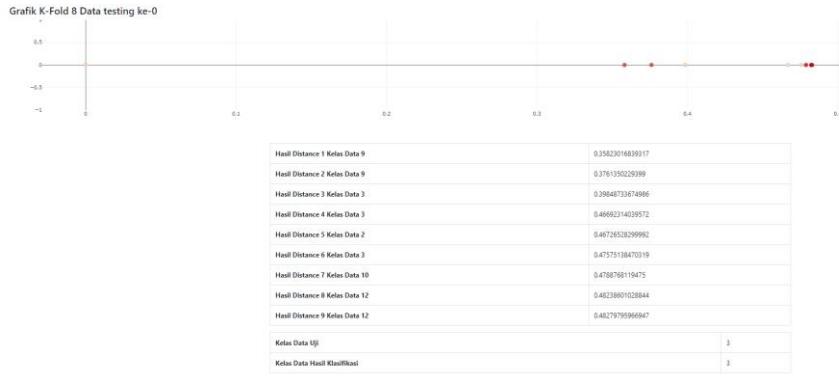


Gambar 10. Output Klasifikasi KNN K-Fold Iterasi ke-8

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 11, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-9.

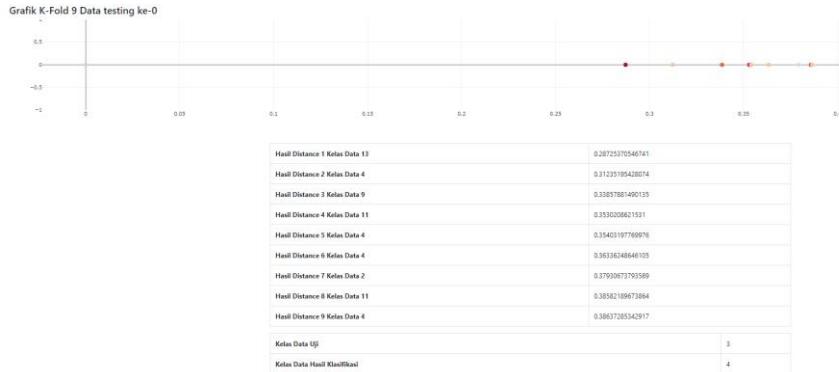
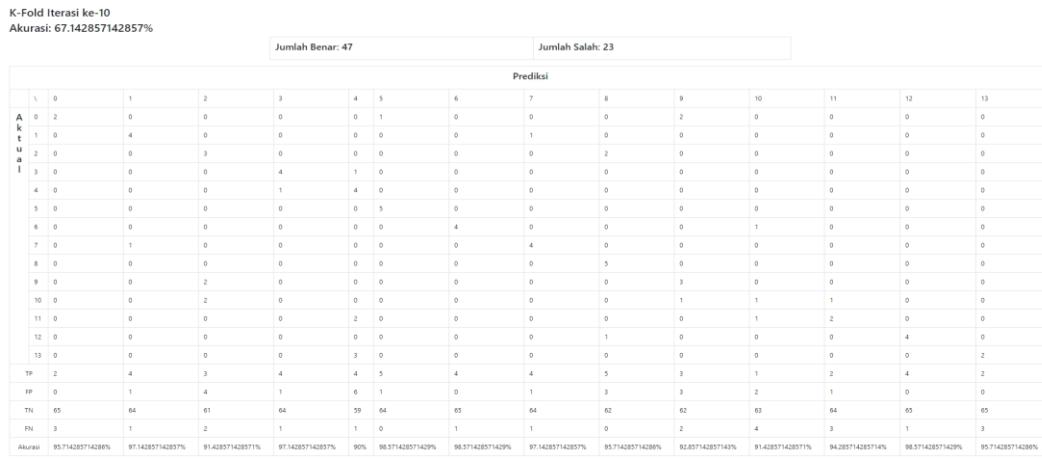
K-Fold Iterasi ke-9  
Akurasi: 60%

Prediksi														
\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	1	0	0	0	0	0	0
k	1	0	4	0	0	0	0	0	0	0	0	0	0	0
t	2	2	0	1	0	1	0	0	0	1	0	0	0	0
u	3	0	0	3	1	0	0	0	0	0	0	0	0	1
a	4	0	0	1	0	2	0	0	0	0	0	0	0	2
i	5	0	0	0	0	0	5	0	0	0	0	0	0	0
1	6	0	0	1	0	0	0	4	0	0	0	0	0	0
3	7	0	0	0	0	0	0	0	5	0	0	0	0	0
0	8	0	0	0	0	0	0	0	5	0	0	0	0	0
1	9	2	0	0	0	0	0	0	0	2	0	0	1	0
0	10	0	0	0	0	1	0	1	0	2	1	0	0	0
1	11	0	0	1	0	1	0	0	0	0	1	0	0	2
0	12	0	0	1	0	0	0	0	0	0	0	0	4	0
1	13	0	0	1	0	0	0	0	0	0	0	2	0	2
TP	4	4	1	3	2	5	4	5	5	2	1	0	4	2
FP	4	0	2	3	4	0	1	2	0	3	1	2	1	5
TN	61	65	63	62	61	65	64	63	65	62	64	63	64	60
FN	1	1	4	2	2	0	1	0	0	3	4	1	3	2
Akurasi	92.857142857143%	98.571428571429%	91.428571428571%	92.857142857143%	90%	100%	97.142857142857%	97.142857142857%	100%	91.428571428571%	92.857142857143%	90%	97.142857142857%	88.571428571429%



Gambar 11. Output Klasifikasi KNN K-Fold Iterasi ke-9

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 12, merupakan hasil dari klasifikasi KNN K-Fold Iterasi ke-10.



Gambar 12. Output Klasifikasi KNN K-Fold Iterasi ke-10

Output dari KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 13, merupakan hasil dari rata–rata akurasi semua K-Fold klasifikasi KNN.

Total Benar: 425  
 Total Salah: 275  
 Total Rata-Rata Akurasi Semua k-fold: 60.714285714286%

Gambar 13. Output Rata–Rata Akurasi Semua K-Fold Klasifikasi KNN

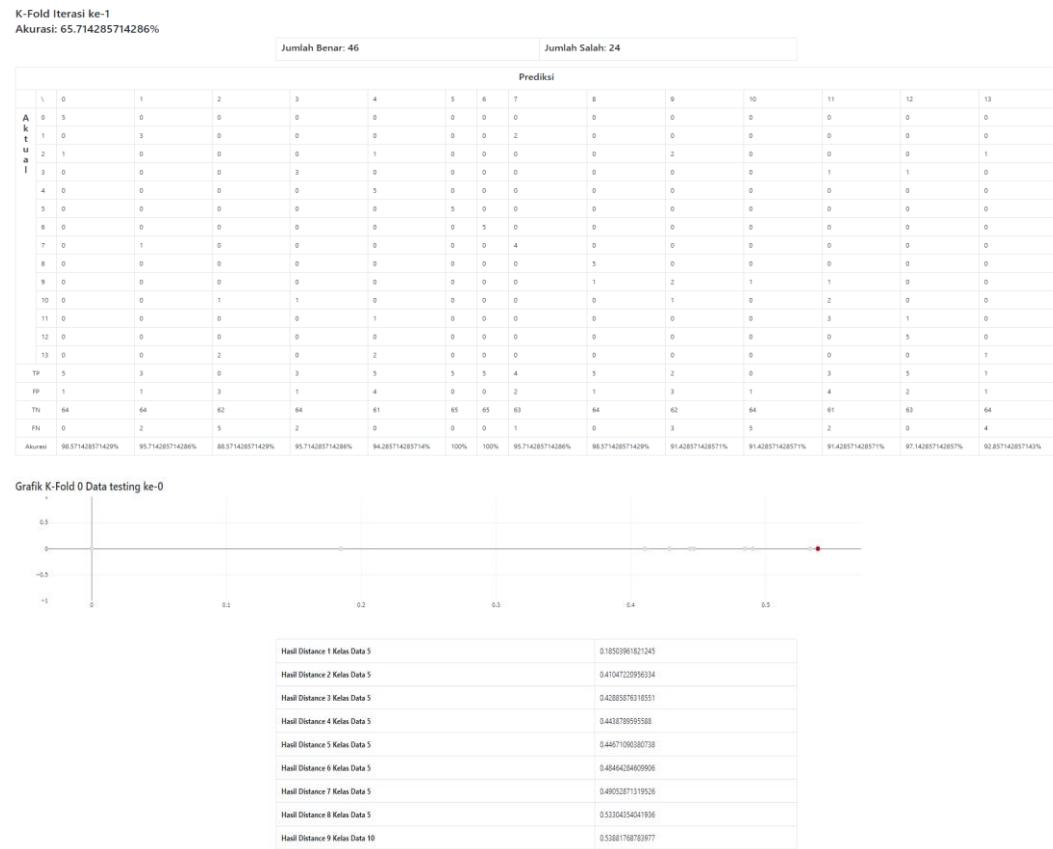
### 3. Output Klasifikasi MKNN

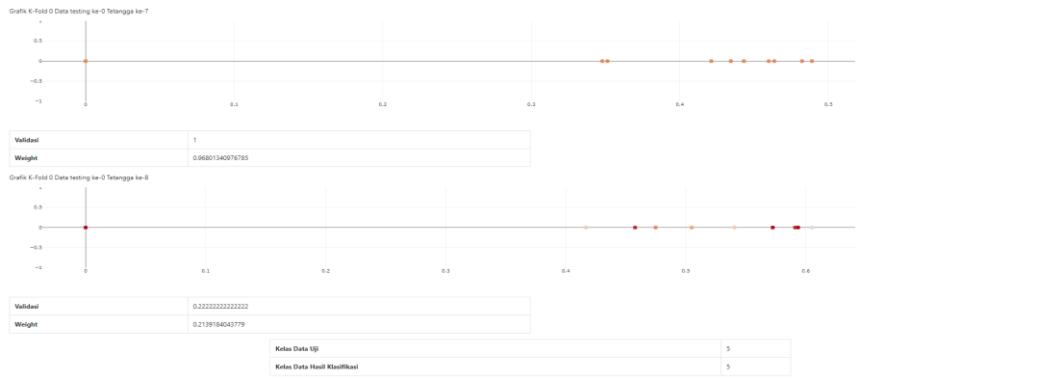
Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 14, merupakan hasil dari klasifikasi MKNN.

PROGRAM STUDI INFORMATIKA														
KLASIFIKASI MUSIK BERDASARKAN GENRE PADA LAYANAN STREAMING MUSIK SPOTIFY MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR DAN MODIFIED K-NEAREST NEIGHBOR														
TANPA SELEKSI FITUR PRINCIPAL COMPONENT ANALYSIS (PCA)														
Data Setelah Normalisasi														
No	Genre	Playlist	Danceability	Energy	Key	Loudness	Mode	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo	Duration
1	Gospel	5	0.541	0.675	0.0909091	0.080785	1	0.066	0.1	0	0.977	0.355	0.425518	0.429394
2	Reggae	11	0.766	0.49	0.545455	0.719925	1	0.3	0.359	0.0085	0.646	0.959	0.0819122	0.137281
3	Hip Hop	6	0.479	0.549	1	0.780909	0	0.373	0.57	0.0239	0.127	0.576	0.783917	0.303463
4	Edm	3	0.873	0.647	0.0909091	0.086668	1	0.103	0.00715	0.0013	0.802	0.385	0.401213	0.416323
5	Country	2	0.546	0.723	0.909091	0.038866	0	0.0394	0.429	0.0232	0.157	0.531	0.321975	0.263891
6	Hip Hop	6	0.917	0.516	0.727273	0.070608	1	0.419	0.149	0	0.092	0.575	0.516089	0.16991
7	Soul	13	0.656	0.32	0.818182	0.730154	1	0.0285	0.521	0.000294	0.159	0.64	0.0949475	0.17982
8	Pop	9	0.496	0.797	0.181818	0.890234	1	0.0379	0.0433	0.0000146	0.155	0.376	0.751388	0.362783
9	Blues	0	0.502	0.112	1	0.801117	1	0.0427	0.877	0.0221	0.104	0.272	0.482047	0.208825
10	Rock	12	0.541	0.968	0	0.80257	1	0.0396	0.000348	0.00000828	0.423	0.505	0.42074	0.383754

Gambar 14. Output Klasifikasi MKNN

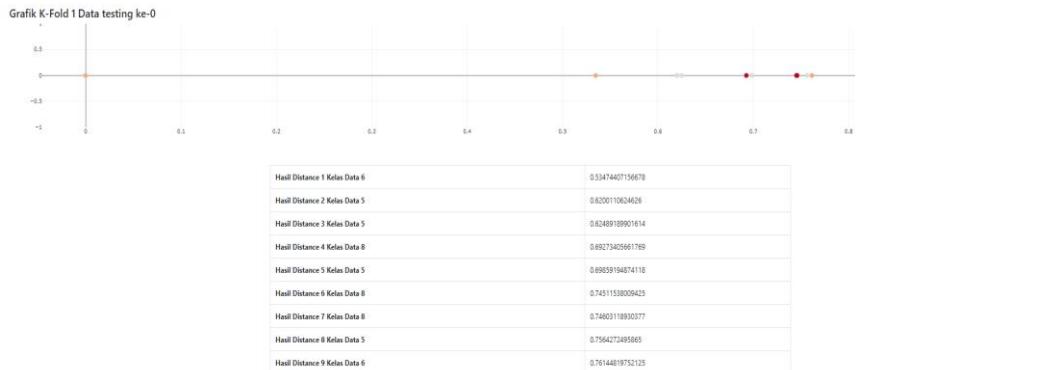
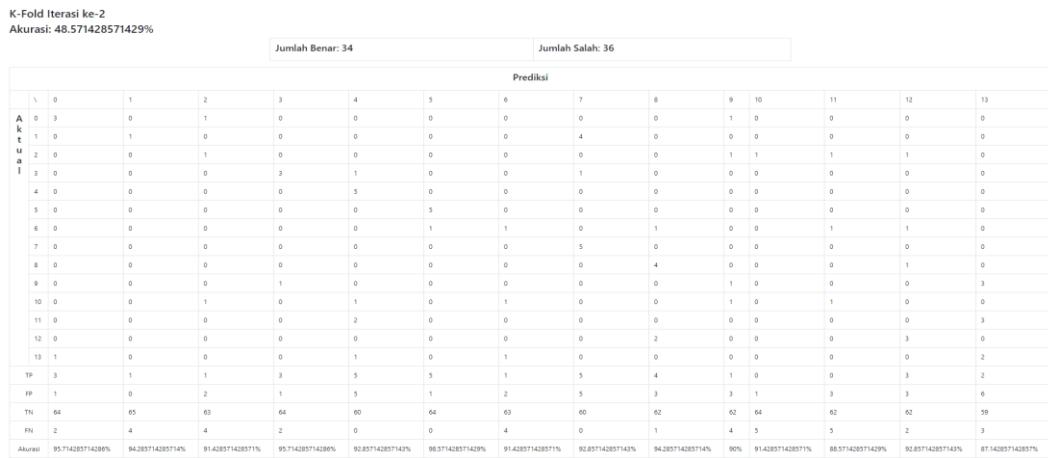
Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 15, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-1.

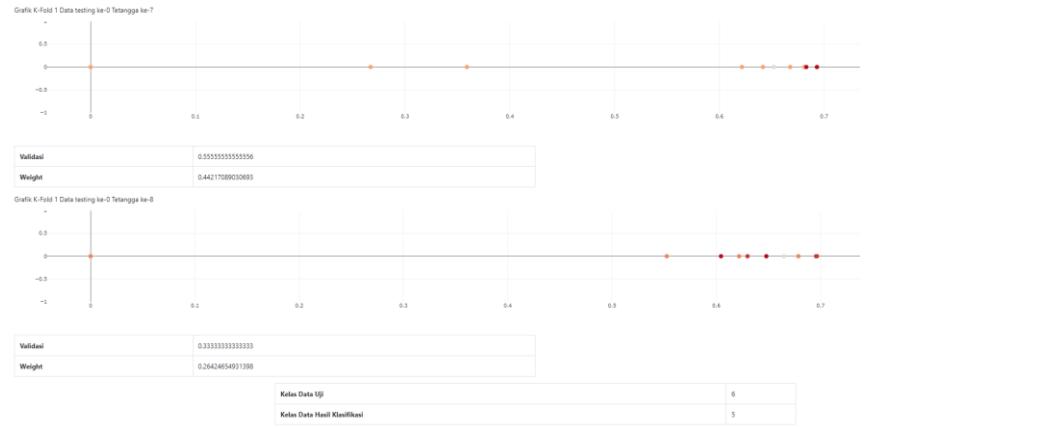




Gambar 15. Output Klasifikasi MKNN K-Fold Iterasi ke-1

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 16, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-2.

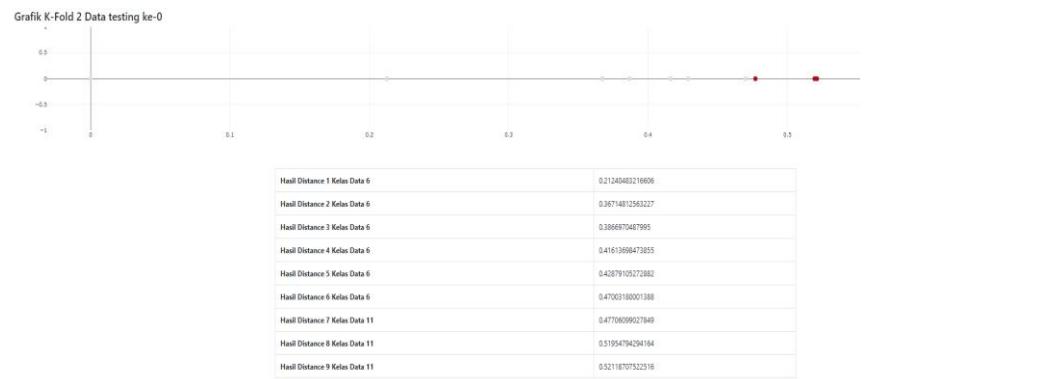


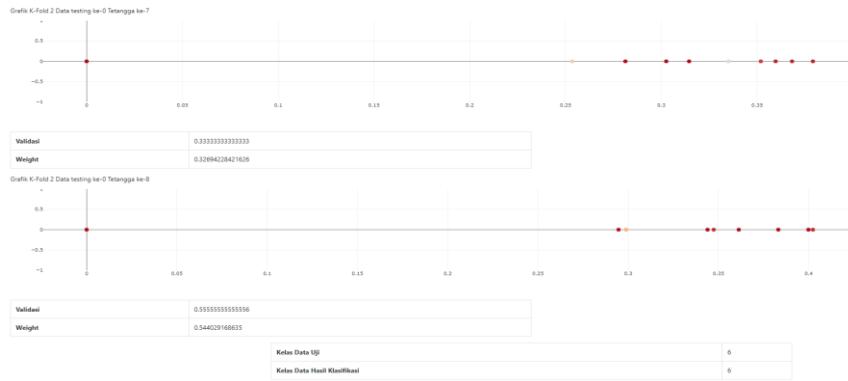


Gambar 16. Output Klasifikasi MKNN K-Fold Iterasi ke-2

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 17, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-3.

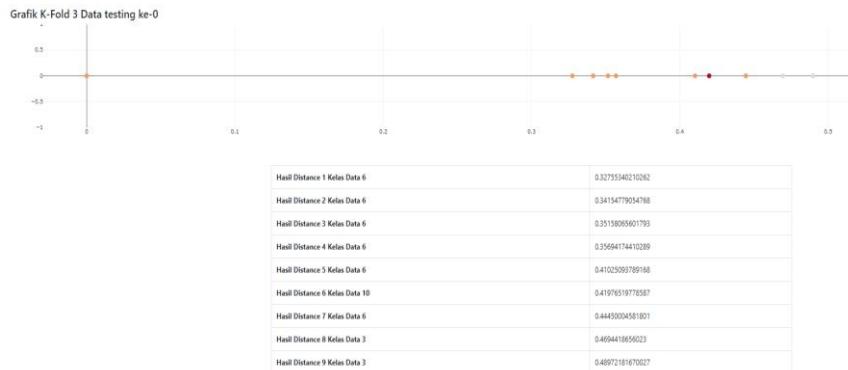
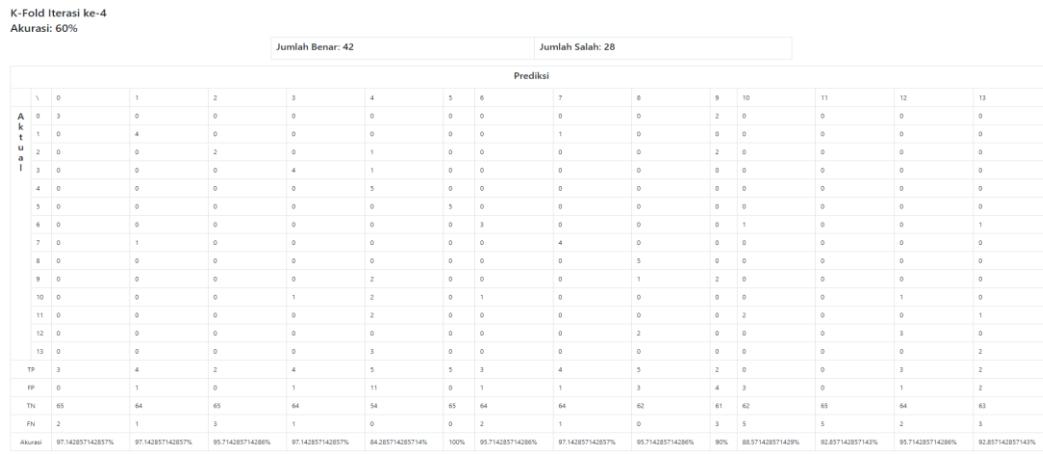
K-Fold Iterasi ke-3														
Akurasi: 60%														
Jumlah Benar: 42 Jumlah Salah: 28														
Prediksi														
\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	0	0	0	1	0
k	1	0	4	0	0	0	0	0	1	0	0	0	0	0
t	2	0	0	1	0	0	0	0	0	0	1	1	1	1
u	3	0	0	0	4	0	0	0	0	0	1	0	0	0
a	4	0	0	0	0	3	0	0	0	0	0	0	0	0
l	5	0	0	0	0	0	5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	4	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	5	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	5	0	0	0	0
9	0	0	3	1	0	0	0	0	0	0	1	0	0	0
10	0	0	1	0	1	0	0	0	0	0	1	0	2	0
11	0	0	1	1	1	0	0	0	0	0	1	1	0	0
12	0	0	0	0	0	0	0	0	3	0	0	0	2	0
13	1	0	1	0	1	0	0	0	0	0	1	0	0	1
TP	4	4	1	4	5	5	4	5	5	0	1	1	2	1
FP	1	0	6	2	3	0	0	1	3	2	3	1	5	1
TN	64	65	59	63	62	65	65	64	62	63	62	64	60	64
FN	1	1	4	1	0	0	1	0	0	5	4	4	3	4
Akurasi	97.142857142857%	98.571428571429%	85.714285714286%	95.714285714286%	100%	98.571428571429%	98.571428571429%	95.714285714286%	90%	90%	92.857142857143%	88.571428571429%	92.857142857143%	

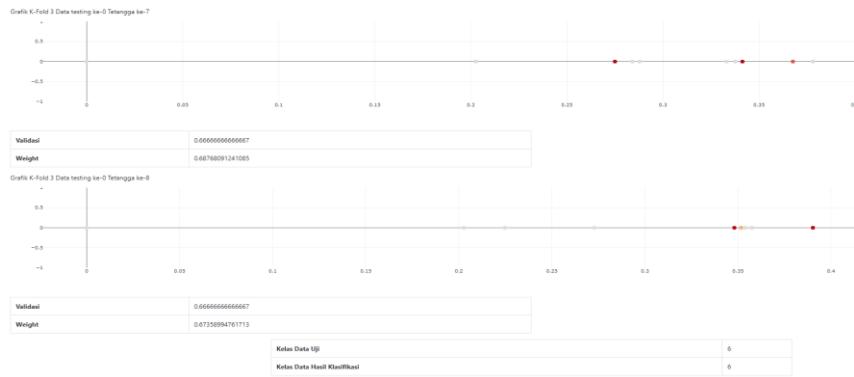




Gambar 17. Output Klasifikasi MKNN K-Fold Iterasi ke-3

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 18, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-4.

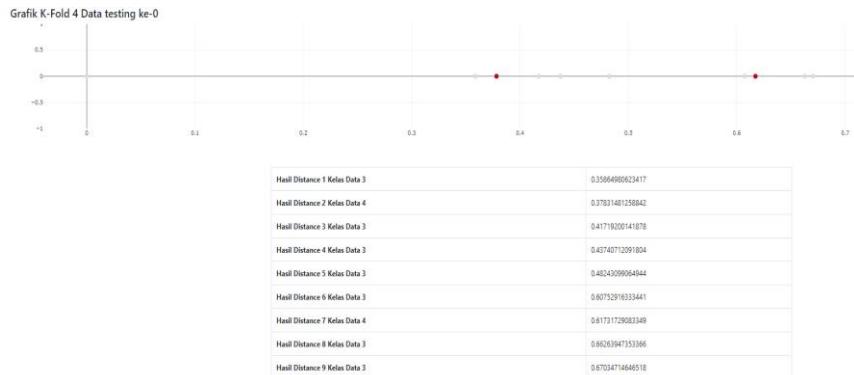


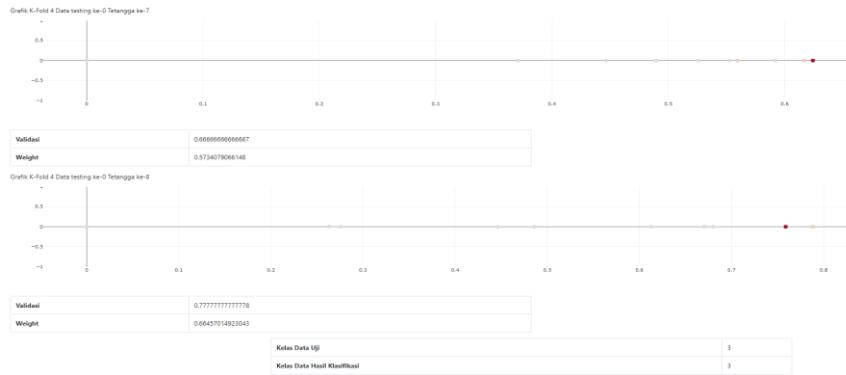


Gambar 18. Output Klasifikasi MKNN K-Fold Iterasi ke-4

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 19, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-5.

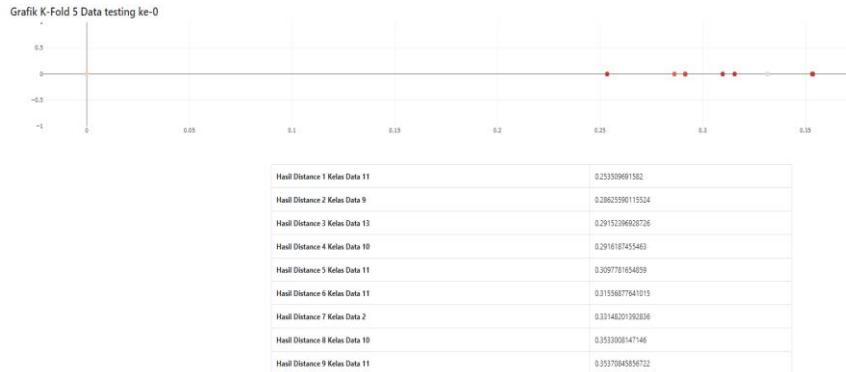
K-Fold Iterasi ke-5														
Akurasi: 60%														
Jumlah Benar: 42 Jumlah Salah: 28														
Prediksi														
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	1	1	0	0	0	0	0	1	0	0	1	0	1
k	1	0	5	0	0	0	0	0	0	0	0	0	0	0
t	2	0	0	1	0	1	0	0	0	1	0	2	0	0
u	3	0	0	0	4	0	0	0	0	0	1	0	0	0
a	4	0	0	0	1	4	0	0	0	0	0	0	0	0
i	5	0	0	0	0	5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	3	0	0	0	0	1	1	0
7	0	1	0	0	0	0	0	0	4	0	0	0	0	0
8	0	0	0	1	0	0	0	0	4	0	0	0	0	0
9	1	0	0	0	0	1	0	0	0	2	0	0	1	0
10	1	0	0	0	0	1	0	0	0	0	2	1	0	0
11	0	0	0	1	1	0	0	0	0	0	0	2	0	1
12	0	0	0	0	0	0	0	0	0	0	0	0	5	0
13	0	0	0	0	0	2	0	0	0	0	2	0	1	0
TP	1	5	1	4	4	5	3	4	4	2	2	2	5	0
FP	2	2	0	3	6	0	0	0	0	2	3	4	4	2
TN	63	63	65	62	59	65	63	65	65	63	62	61	61	63
FN	4	0	4	1	1	0	2	1	1	3	3	0	5	0
Akurasi	91.428571428571%	97.1428571428571%	94.285714285714%	94.285714285714%	90%	100%	97.142857142857%	98.571428571429%	98.57142857143%	92.857142857143%	91.428571428571%	90%	94.285714285714%	90%

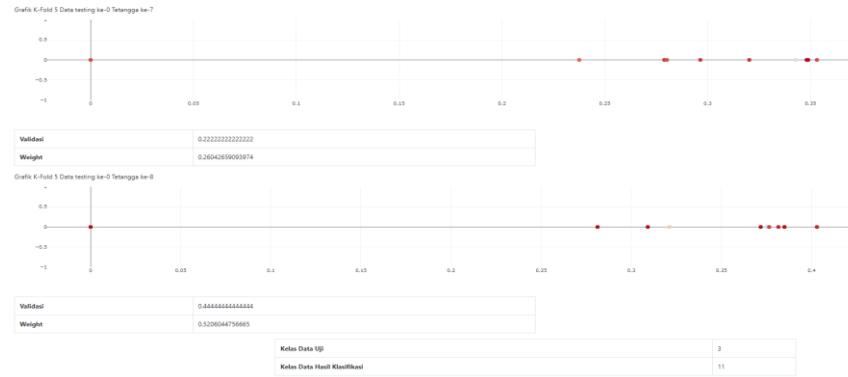




Gambar 19. Output Klasifikasi MKNN K-Fold Iterasi ke-5

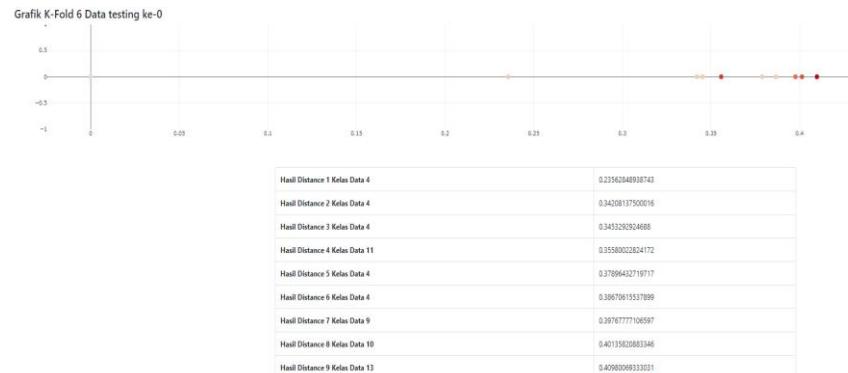
Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 20, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-6.

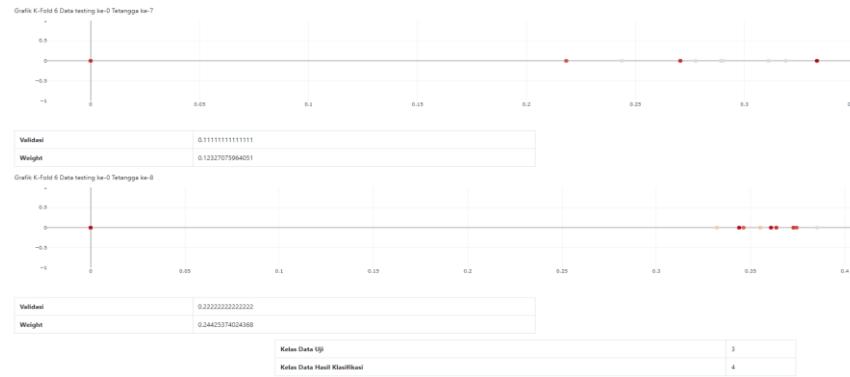




Gambar 20. Output Klasifikasi MKNN K-Fold Iterasi ke-6

Output dari MKNN akan munjur jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 21, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-7.

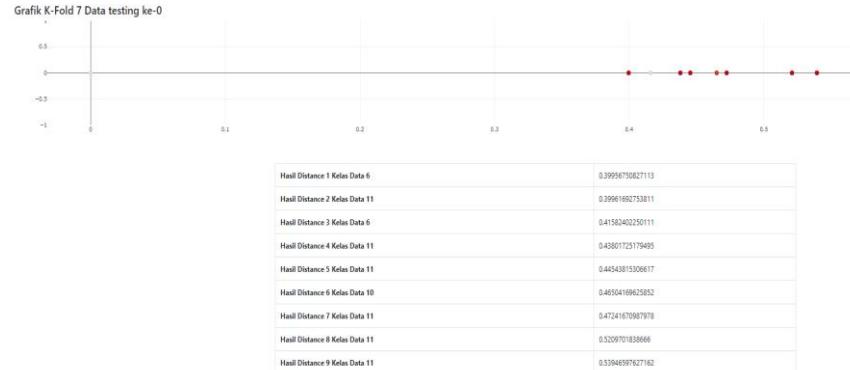


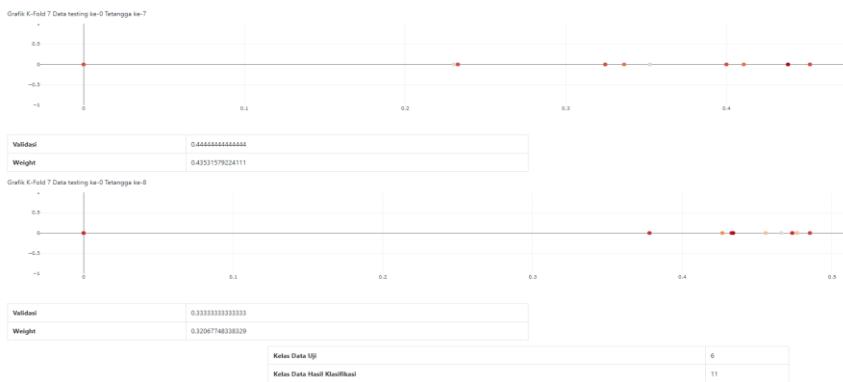


Gambar 21. Output Klasifikasi MKNN K-Fold Iterasi ke-7

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 22, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-8.

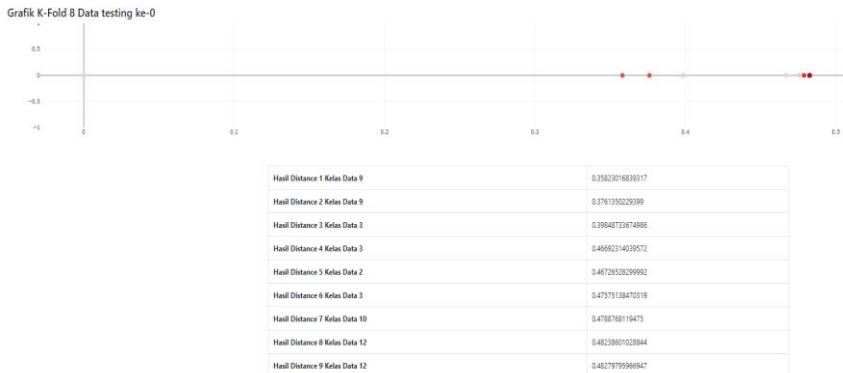
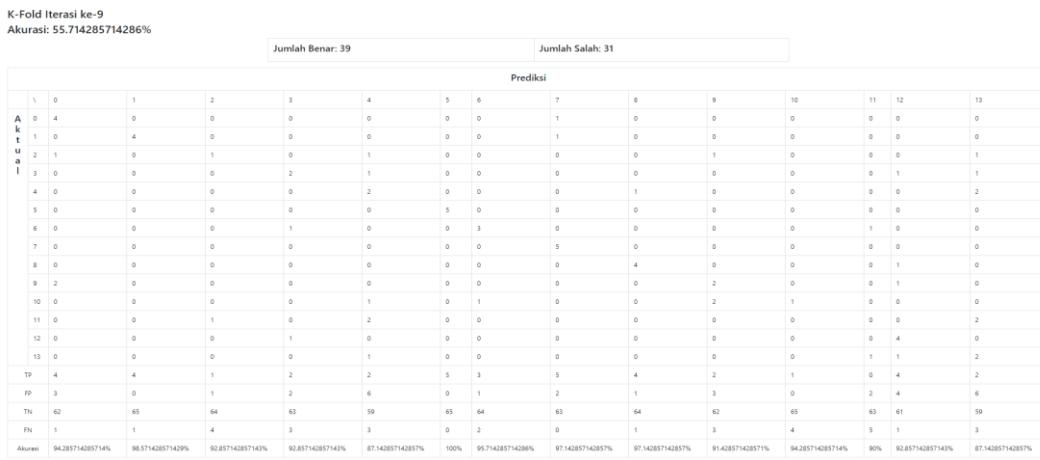
Prediksi														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	0	0	1	0	0
k	1	0	0	0	0	0	0	5	0	0	0	0	0	0
t	2	1	0	2	0	0	0	0	0	1	0	0	1	0
u	3	0	0	0	4	0	0	0	0	0	0	0	1	0
a	4	0	0	0	4	0	0	0	0	0	0	0	0	1
i	5	0	0	0	0	4	0	0	0	0	0	0	1	0
l	6	0	0	0	0	0	3	0	0	0	0	1	1	0
1	7	0	1	0	0	0	0	4	0	0	0	0	0	0
8	8	0	0	0	0	0	0	0	4	0	0	0	1	0
9	9	1	0	0	1	0	0	0	0	3	0	0	0	0
10	10	0	0	0	1	1	0	0	0	0	0	1	1	1
11	11	0	0	1	0	1	0	0	0	0	0	0	3	0
12	12	0	0	0	1	0	0	0	0	0	0	0	3	0
13	13	1	0	0	1	3	0	0	0	0	0	0	0	0
TP	4	0	2	4	4	4	3	4	4	3	0	3	3	0
FP	3	1	1	4	5	0	0	5	0	1	1	3	6	2
TN	62	64	64	61	60	65	65	60	65	64	64	62	59	63
FN	1	5	3	1	1	1	2	1	1	2	5	2	2	5
Akurasi	94.285714285714%	91.428571428571%	94.285714285714%	92.857142857143%	91.428571428571%	98.571428571428%	97.142857142857%	91.428571428571%	98.571428571428%	91.42857142857143%	92.8571428571429%	88.571428571429%	90%	

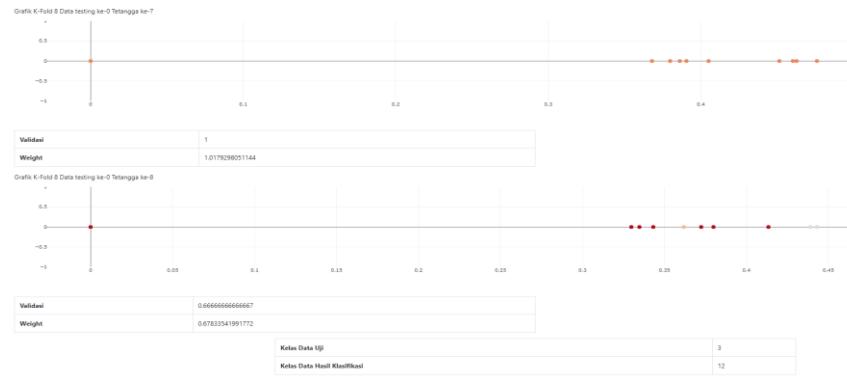




Gambar 22. Output Klasifikasi MKNN K-Fold Iterasi ke-8

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 23, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-9.

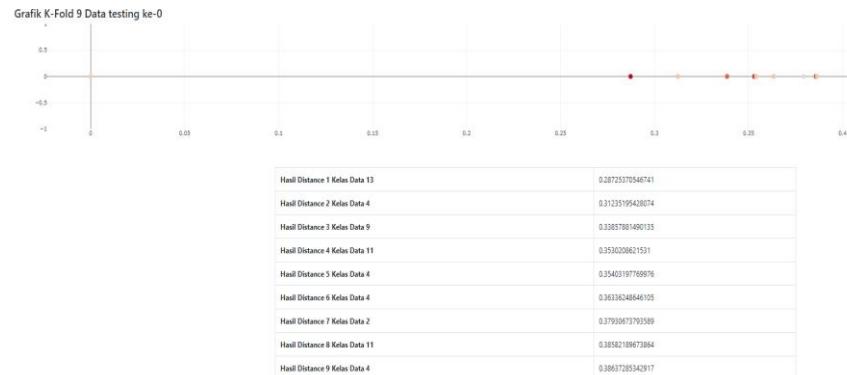


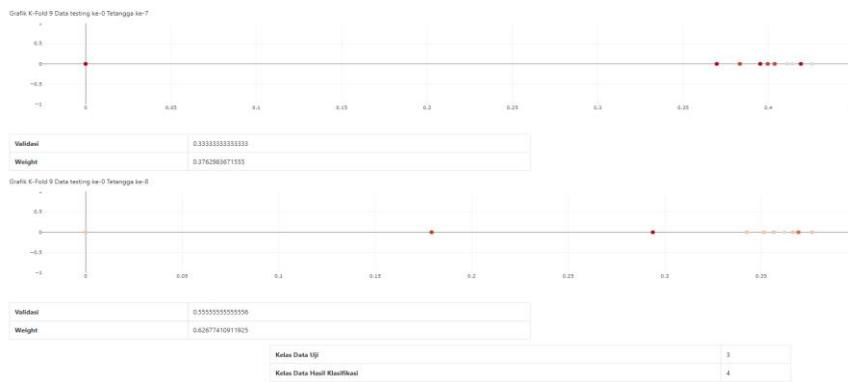


Gambar 23. Output Klasifikasi MKNN K-Fold Iterasi ke-9

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 24, merupakan hasil dari klasifikasi MKNN K-Fold Iterasi ke-10.

Prediksi																
Jumlah Benar: 50 Jumlah Salah: 20																
A	0	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13
k	1	0	5	0	0	0	0	0	0	0	2	0	0	0	0	0
t	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1
u	3	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0
a	4	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0
a	5	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
l	6	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0	4	0	0	0	0	0
10	0	0	1	0	1	0	0	0	0	0	0	1	1	1	1	0
11	0	0	0	0	2	0	0	0	0	0	0	0	2	0	0	1
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0
13	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	3
TP	2	5	2	4	4	5	5	4	5	4	1	2	4	3		
FP	0	1	2	1	6	1	0	0	3	2	0	1	1	2		
TN	65	64	63	64	59	64	65	65	62	63	65	64	64	63		
FN	3	0	3	1	1	0	0	1	0	1	4	3	1	2		
Akurasi	95.714285714285%	98.571428571429%	92.857142857143%	97.142857142857%	90%	98.571428571429%	100%	98.571428571429%	95.714285714286%	94.285714285714%	94.285714285714%	97.142857142857%	94.285714285714%			





Gambar 24. Output Klasifikasi MKNN K-Fold Iterasi ke-10

Output dari MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 25, merupakan hasil dari rata–rata akurasi semua K-Fold klasifikasi MKNN.

Total Benar: 393  
 Total Salah: 307  
 Total Rata-Rata Akurasi Semua k-fold: 56.142857142857%

Gambar 25. Output Rata–Rata Akurasi Semua K-Fold Klasifikasi MKNN

## LAMPIRAN C

### Pengujian Kedua

#### Lampiran C.1. Data Seleksi Fitur menggunakan *Principal Component Analysis* (PCA)

##### 1. Reduksi menggunakan PCA dengan 1 variabel fitur baru.

no	playlist	pca1
1	5	0.231265
2	11	0.019342
3	6	0.083597
4	3	0.327128
5	2	0.155986
6	6	0.213198
7	13	-0.22343
8	9	0.300049
9	0	-0.67088
10	12	0.410438
11	3	-0.25352
12	5	-0.10885
13	12	0.441466
14	5	0.430253
15	5	0.090228

16	0	-0.133
17	3	0.240056
18	7	-0.88528
19	7	-1.1949
20	11	0.348535
21	7	-1.32714
22	13	0.179812
23	1	-1.05089
24	11	0.332005
25	7	-0.81025
26	4	0.397105
27	4	0.335913
28	6	0.23205
29	13	0.157281
30	0	-0.1676
31	9	0.28378
32	10	0.1715
33	13	0.301444
34	10	0.348377
35	6	0.46293
36	6	0.271035
37	5	0.386602
38	4	0.467721
39	6	0.451506
40	1	-1.44525
41	5	-0.007
42	1	-1.29466
43	8	0.484525
44	2	-0.1047
45	5	0.275286
46	7	-1.06863
47	10	0.23167
48	10	0.046785
49	6	0.172692
50	6	0.337794
51	8	-0.00384
52	0	-0.23863
53	11	0.174643
54	12	0.369431
55	6	0.460604
56	11	0.428497
57	11	0.221119
58	4	0.429467

59	2	-0.01772
60	4	0.314618
61	8	0.447487
62	4	0.479009
63	13	-0.2082
64	4	0.510406
65	13	0.072784
66	3	0.515228
67	4	0.378022
68	9	0.079178
69	6	0.43555
70	13	0.307345
71	2	0.099326
72	2	0.619017
73	4	0.261215
74	9	0.420581
75	13	0.044186
76	9	-0.41291
77	3	0.114251
78	2	0.149082
79	13	-0.12791
80	1	-0.87763
81	8	0.347725
82	9	-0.27963
83	9	0.187446
84	3	0.244012
85	10	-0.05832
86	9	0.343678
87	3	-0.53178
88	12	0.418114
89	9	-0.08109
90	6	0.14391
91	13	0.014701
92	12	0.508023
93	0	-0.35717
94	12	0.391258
95	6	0.33457
96	7	-1.10065
97	4	0.470013
98	9	0.197938
99	3	0.494559
100	1	-1.26741
-	-	-

-	-	-	-
-	-	-	-
-	-	-	-
700	7	7	-0.9879

## 2. Reduksi menggunakan PCA dengan 2 variabel fitur baru.

no	playlist	pca1	pca2
1	5	0.231265	-0.48271
2	11	0.019342	-0.37475
3	6	0.083597	0.684155
4	3	0.327128	-0.48119
5	2	0.155986	0.66781
6	6	0.213198	-0.37235
7	13	-0.22343	-0.29291
8	9	0.300049	-0.47058
9	0	-0.67088	-0.19968
10	12	0.410438	-0.51446
11	3	-0.25352	-0.41067
12	5	-0.10885	-0.35925
13	12	0.441466	0.548259
14	5	0.430253	-0.51853
15	5	0.090228	-0.35081
16	0	-0.133	0.572266
17	3	0.240056	-0.38394
18	7	-0.88528	0.742997
19	7	-1.1949	-0.27278
20	11	0.348535	-0.38023
21	7	-1.32714	-0.16145
22	13	0.179812	-0.45495
23	1	-1.05089	-0.22557
24	11	0.332005	0.577097
25	7	-0.81025	-0.26785
26	4	0.397105	-0.3818
27	4	0.335913	-0.51065
28	6	0.23205	-0.41634
29	13	0.157281	-0.36342
30	0	-0.1676	0.728406
31	9	0.28378	-0.39756
32	10	0.1715	-0.37293
33	13	0.301444	-0.40455
34	10	0.348377	0.560023
35	6	0.46293	0.627604

36	6	0.271035	-0.38152
37	5	0.386602	-0.51347
38	4	0.467721	0.614639
39	6	0.451506	0.620517
40	1	-1.44525	-0.1909
41	5	-0.007	-0.36886
42	1	-1.29466	-0.19152
43	8	0.484525	-0.52136
44	2	-0.1047	0.664928
45	5	0.275286	-0.4461
46	7	-1.06863	0.771032
47	10	0.23167	-0.39483
48	10	0.046785	0.599216
49	6	0.172692	0.61368
50	6	0.337794	-0.38823
51	8	-0.00384	-0.32722
52	0	-0.23863	0.688637
53	11	0.174643	-0.37931
54	12	0.369431	-0.36222
55	6	0.460604	0.511221
56	11	0.428497	0.566894
57	11	0.221119	-0.43318
58	4	0.429467	0.632907
59	2	-0.01772	-0.34697
60	4	0.314618	-0.49886
61	8	0.447487	-0.49909
62	4	0.479009	0.532916
63	13	-0.2082	-0.39863
64	4	0.510406	0.504206
65	13	0.072784	0.652389
66	3	0.515228	0.52658
67	4	0.378022	-0.3543
68	9	0.079178	-0.43491
69	6	0.43555	-0.41472
70	13	0.307345	-0.49864
71	2	0.099326	-0.43436
72	2	0.619017	0.500245
73	4	0.261215	-0.399
74	9	0.420581	0.634775
75	13	0.044186	0.558131
76	9	-0.41291	-0.33036
77	3	0.114251	-0.45985
78	2	0.149082	-0.38803

79	13	-0.12791	-0.42984
80	1	-0.87763	0.764416
81	8	0.347725	-0.39723
82	9	-0.27963	-0.30388
83	9	0.187446	0.55789
84	3	0.244012	-0.39007
85	10	-0.05832	0.598124
86	9	0.343678	0.640593
87	3	-0.53178	0.648765
88	12	0.418114	-0.43876
89	9	-0.08109	-0.2912
90	6	0.14391	-0.31283
91	13	0.014701	-0.45613
92	12	0.508023	0.610559
93	0	-0.35717	0.748595
94	12	0.391258	-0.42018
95	6	0.33457	-0.38257
96	7	-1.10065	0.738401
97	4	0.470013	0.627129
98	9	0.197938	0.548307
99	3	0.494559	-0.44128
100	1	-1.26741	-0.14138
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
700	7	-0.9879	0.795137

### 3. Reduksi menggunakan PCA dengan 3 variabel fitur baru.

no	playlist	pca1	pca2	pca3
1	5	0.231265	-0.48271	-0.52019
2	11	0.019342	-0.37475	0.575705
3	6	0.083597	0.684155	0.202254
4	3	0.327128	-0.48119	-0.18497
5	2	0.155986	0.66781	0.239623
6	6	0.213198	-0.37235	0.352747
7	13	-0.22343	-0.29291	0.545976
8	9	0.300049	-0.47058	-0.39231
9	0	-0.67088	-0.19968	0.393497
10	12	0.410438	-0.51446	-0.4221
11	3	-0.25352	-0.41067	-0.2665
12	5	-0.10885	-0.35925	-0.16653

13	12	0.441466	0.548259	-0.2964
14	5	0.430253	-0.51853	-0.59208
15	5	0.090228	-0.35081	-0.15948
16	0	-0.133	0.572266	-0.2688
17	3	0.240056	-0.38394	0.094423
18	7	-0.88528	0.742997	-0.33648
19	7	-1.1949	-0.27278	-0.33955
20	11	0.348535	-0.38023	0.460184
21	7	-1.32714	-0.16145	0.076057
22	13	0.179812	-0.45495	0.192115
23	1	-1.05089	-0.22557	0.199849
24	11	0.332005	0.577097	-0.02851
25	7	-0.81025	-0.26785	0.265505
26	4	0.397105	-0.3818	0.529941
27	4	0.335913	-0.51065	-0.00078
28	6	0.23205	-0.41634	0.117852
29	13	0.157281	-0.36342	0.511434
30	0	-0.1676	0.728406	0.289961
31	9	0.28378	-0.39756	0.301136
32	10	0.1715	-0.37293	0.215676
33	13	0.301444	-0.40455	0.388238
34	10	0.348377	0.560023	0.285323
35	6	0.46293	0.627604	-0.0993
36	6	0.271035	-0.38152	0.045862
37	5	0.386602	-0.51347	-0.49744
38	4	0.467721	0.614639	0.570252
39	6	0.451506	0.620517	-0.34683
40	1	-1.44525	-0.1909	-0.13045
41	5	-0.007	-0.36886	-0.34799
42	1	-1.29466	-0.19152	-0.36372
43	8	0.484525	-0.52136	-0.61169
44	2	-0.1047	0.664928	0.503343
45	5	0.275286	-0.4461	-0.62836
46	7	-1.06863	0.771032	-0.13556
47	10	0.23167	-0.39483	0.378084
48	10	0.046785	0.599216	-0.08031
49	6	0.172692	0.61368	-0.00939
50	6	0.337794	-0.38823	0.493163
51	8	-0.00384	-0.32722	-0.4843
52	0	-0.23863	0.688637	0.159441
53	11	0.174643	-0.37931	0.148202
54	12	0.369431	-0.36222	-0.10193
55	6	0.460604	0.511221	-0.03738

56	11	0.428497	0.566894	0.300943
57	11	0.221119	-0.43318	0.287604
58	4	0.429467	0.632907	0.403308
59	2	-0.01772	-0.34697	0.099972
60	4	0.314618	-0.49886	0.06095
61	8	0.447487	-0.49909	-0.51014
62	4	0.479009	0.532916	0.143784
63	13	-0.2082	-0.39863	0.110611
64	4	0.510406	0.504206	0.089037
65	13	0.072784	0.652389	0.228934
66	3	0.515228	0.52658	-0.06786
67	4	0.378022	-0.3543	0.631394
68	9	0.079178	-0.43491	-0.22581
69	6	0.43555	-0.41472	-0.18276
70	13	0.307345	-0.49864	0.112861
71	2	0.099326	-0.43436	-0.01656
72	2	0.619017	0.500245	0.153352
73	4	0.261215	-0.399	0.545679
74	9	0.420581	0.634775	0.240972
75	13	0.044186	0.558131	-0.07237
76	9	-0.41291	-0.33036	-0.09738
77	3	0.114251	-0.45985	-0.13403
78	2	0.149082	-0.38803	0.230113
79	13	-0.12791	-0.42984	0.142717
80	1	-0.87763	0.764416	0.076287
81	8	0.347725	-0.39723	-0.35673
82	9	-0.27963	-0.30388	0.072852
83	9	0.187446	0.55789	0.27587
84	3	0.244012	-0.39007	0.203124
85	10	-0.05832	0.598124	-0.16713
86	9	0.343678	0.640593	0.173923
87	3	-0.53178	0.648765	0.24916
88	12	0.418114	-0.43876	-0.32308
89	9	-0.08109	-0.2912	0.536306
90	6	0.14391	-0.31283	0.373934
91	13	0.014701	-0.45613	0.207174
92	12	0.508023	0.610559	-0.20592
93	0	-0.35717	0.748595	0.160763
94	12	0.391258	-0.42018	-0.27113
95	6	0.33457	-0.38257	0.41042
96	7	-1.10065	0.738401	-0.29283
97	4	0.470013	0.627129	0.559982
98	9	0.197938	0.548307	-0.26877

99	3	0.494559	-0.44128	0.365255
100	1	-1.26741	-0.14138	0.209059
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
700	7	-0.9879	0.795137	-0.33482

#### 4. Reduksi menggunakan PCA dengan 4 variabel fitur baru.

no	playlist	pca1	pca2	pca3	pca4
1	5	0.231265	-0.48271	-0.52019	0.012135
2	11	0.019342	-0.37475	0.575705	-0.24897
3	6	0.083597	0.684155	0.202254	0.376243
4	3	0.327128	-0.48119	-0.18497	-0.30859
5	2	0.155986	0.66781	0.239623	0.224091
6	6	0.213198	-0.37235	0.352747	0.106779
7	13	-0.22343	-0.29291	0.545976	0.120568
8	9	0.300049	-0.47058	-0.39231	-0.03513
9	0	-0.67088	-0.19968	0.393497	0.473913
10	12	0.410438	-0.51446	-0.4221	-0.22267
11	3	-0.25352	-0.41067	-0.2665	-0.3842
12	5	-0.10885	-0.35925	-0.16653	0.33659
13	12	0.441466	0.548259	-0.2964	-0.06949
14	5	0.430253	-0.51853	-0.59208	0.056193
15	5	0.090228	-0.35081	-0.15948	0.525331
16	0	-0.133	0.572266	-0.2688	-0.50954
17	3	0.240056	-0.38394	0.094423	0.162781
18	7	-0.88528	0.742997	-0.33648	0.086093
19	7	-1.1949	-0.27278	-0.33955	-0.26624
20	11	0.348535	-0.38023	0.460184	0.131019
21	7	-1.32714	-0.16145	0.076057	0.115466
22	13	0.179812	-0.45495	0.192115	-0.44353
23	1	-1.05089	-0.22557	0.199849	-0.10297
24	11	0.332005	0.577097	-0.02851	-0.15625
25	7	-0.81025	-0.26785	0.265505	-0.15692
26	4	0.397105	-0.3818	0.529941	0.072798
27	4	0.335913	-0.51065	-0.00078	-0.54069
28	6	0.23205	-0.41634	0.117852	-0.05667
29	13	0.157281	-0.36342	0.511434	0.005603
30	0	-0.1676	0.728406	0.289961	0.317466
31	9	0.28378	-0.39756	0.301136	0.026411
32	10	0.1715	-0.37293	0.215676	0.20215

33	13	0.301444	-0.40455	0.388238	-0.01701
34	10	0.348377	0.560023	0.285323	-0.37339
35	6	0.46293	0.627604	-0.0993	0.452122
36	6	0.271035	-0.38152	0.045862	0.320106
37	5	0.386602	-0.51347	-0.49744	-0.01154
38	4	0.467721	0.614639	0.570252	-0.03316
39	6	0.451506	0.620517	-0.34683	0.665153
40	1	-1.44525	-0.1909	-0.13045	-0.07316
41	5	-0.007	-0.36886	-0.34799	0.454562
42	1	-1.29466	-0.19152	-0.36372	0.19482
43	8	0.484525	-0.52136	-0.61169	0.030867
44	2	-0.1047	0.664928	0.503343	-0.15131
45	5	0.275286	-0.4461	-0.62836	0.293535
46	7	-1.06863	0.771032	-0.13556	-0.05265
47	10	0.23167	-0.39483	0.378084	-0.01674
48	10	0.046785	0.599216	-0.08031	-0.20669
49	6	0.172692	0.61368	-0.00939	-0.03567
50	6	0.337794	-0.38823	0.493163	0.1217
51	8	-0.00384	-0.32722	-0.4843	0.650652
52	0	-0.23863	0.688637	0.159441	-0.02287
53	11	0.174643	-0.37931	0.148202	0.149029
54	12	0.369431	-0.36222	-0.10193	0.532704
55	6	0.460604	0.511221	-0.03738	-0.35881
56	11	0.428497	0.566894	0.300943	-0.27028
57	11	0.221119	-0.43318	0.287604	-0.28488
58	4	0.429467	0.632907	0.403308	0.209772
59	2	-0.01772	-0.34697	0.099972	0.179849
60	4	0.314618	-0.49886	0.06095	-0.57978
61	8	0.447487	-0.49909	-0.51014	0.056886
62	4	0.479009	0.532916	0.143784	-0.25616
63	13	-0.2082	-0.39863	0.110611	-0.35391
64	4	0.510406	0.504206	0.089037	-0.4792
65	13	0.072784	0.652389	0.228934	-0.00754
66	3	0.515228	0.52658	-0.06786	-0.24859
67	4	0.378022	-0.3543	0.631394	0.229037
68	9	0.079178	-0.43491	-0.22581	-0.14197
69	6	0.43555	-0.41472	-0.18276	0.369641
70	13	0.307345	-0.49864	0.112861	-0.48951
71	2	0.099326	-0.43436	-0.01656	-0.16131
72	2	0.619017	0.500245	0.153352	-0.40097
73	4	0.261215	-0.399	0.545679	-0.13994
74	9	0.420581	0.634775	0.240972	0.193651
75	13	0.044186	0.558131	-0.07237	-0.42054

76	9	-0.41291	-0.33036	-0.09738	0.027616
77	3	0.114251	-0.45985	-0.13403	-0.37049
78	2	0.149082	-0.38803	0.230113	-0.02255
79	13	-0.12791	-0.42984	0.142717	-0.48621
80	1	-0.87763	0.764416	0.076287	0.014638
81	8	0.347725	-0.39723	-0.35673	0.469246
82	9	-0.27963	-0.30388	0.072852	0.222388
83	9	0.187446	0.55789	0.27587	-0.45073
84	3	0.244012	-0.39007	0.203124	0.064748
85	10	-0.05832	0.598124	-0.16713	-0.25198
86	9	0.343678	0.640593	0.173923	0.252207
87	3	-0.53178	0.648765	0.24916	-0.54433
88	12	0.418114	-0.43876	-0.32308	0.248481
89	9	-0.08109	-0.2912	0.536306	0.336041
90	6	0.14391	-0.31283	0.373934	0.421317
91	13	0.014701	-0.45613	0.207174	-0.53767
92	12	0.508023	0.610559	-0.20592	0.397478
93	0	-0.35717	0.748595	0.160763	0.424655
94	12	0.391258	-0.42018	-0.27113	0.306223
95	6	0.33457	-0.38257	0.41042	0.195274
96	7	-1.10065	0.738401	-0.29283	-0.25766
97	4	0.470013	0.627129	0.559982	0.088578
98	9	0.197938	0.548307	-0.26877	-0.30945
99	3	0.494559	-0.44128	0.365255	-0.15743
100	1	-1.26741	-0.14138	0.209059	0.228851
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
700	7	-0.9879	0.795137	-0.33482	0.300566

## 5. Reduksi menggunakan PCA dengan 5 variabel fitur baru.

no	playlist	pca1	pca2	pca3	pca4	pca5
1	5	0.231265	-0.48271	-0.52019	0.012135	0.473969
2	11	0.019342	-0.37475	0.575705	-0.24897	0.080159
3	6	0.083597	0.684155	0.202254	0.376243	-0.03749
4	3	0.327128	-0.48119	-0.18497	-0.30859	-0.03803
5	2	0.155986	0.66781	0.239623	0.224091	0.066383
6	6	0.213198	-0.37235	0.352747	0.106779	-0.03699
7	13	-0.22343	-0.29291	0.545976	0.120568	0.273098
8	9	0.300049	-0.47058	-0.39231	-0.03513	-0.11167
9	0	-0.67088	-0.19968	0.393497	0.473913	0.338146

10	12	0.410438	-0.51446	-0.4221	-0.22267	0.059703
11	3	-0.25352	-0.41067	-0.2665	-0.3842	-0.44227
12	5	-0.10885	-0.35925	-0.16653	0.33659	0.66451
13	12	0.441466	0.548259	-0.2964	-0.06949	0.194046
14	5	0.430253	-0.51853	-0.59208	0.056193	0.260659
15	5	0.090228	-0.35081	-0.15948	0.525331	0.571935
16	0	-0.133	0.572266	-0.2688	-0.50954	0.373133
17	3	0.240056	-0.38394	0.094423	0.162781	-0.0135
18	7	-0.88528	0.742997	-0.33648	0.086093	-0.23618
19	7	-1.1949	-0.27278	-0.33955	-0.26624	-0.10043
20	11	0.348535	-0.38023	0.460184	0.131019	-0.107
21	7	-1.32714	-0.16145	0.076057	0.115466	0.01146
22	13	0.179812	-0.45495	0.192115	-0.44353	0.033678
23	1	-1.05089	-0.22557	0.199849	-0.10297	-0.25414
24	11	0.332005	0.577097	-0.02851	-0.15625	0.075497
25	7	-0.81025	-0.26785	0.265505	-0.15692	-0.31253
26	4	0.397105	-0.3818	0.529941	0.072798	-0.16578
27	4	0.335913	-0.51065	-0.00078	-0.54069	-0.21645
28	6	0.23205	-0.41634	0.117852	-0.05667	-0.02571
29	13	0.157281	-0.36342	0.511434	0.005603	0.049549
30	0	-0.1676	0.728406	0.289961	0.317466	0.232327
31	9	0.28378	-0.39756	0.301136	0.026411	-0.04405
32	10	0.1715	-0.37293	0.215676	0.20215	-0.0982
33	13	0.301444	-0.40455	0.388238	-0.01701	0.087048
34	10	0.348377	0.560023	0.285323	-0.37339	0.036937
35	6	0.46293	0.627604	-0.0993	0.452122	0.046175
36	6	0.271035	-0.38152	0.045862	0.320106	-0.14454
37	5	0.386602	-0.51347	-0.49744	-0.01154	0.335967
38	4	0.467721	0.614639	0.570252	-0.03316	-0.10372
39	6	0.451506	0.620517	-0.34683	0.665153	0.265388
40	1	-1.44525	-0.1909	-0.13045	-0.07316	-0.01291
41	5	-0.007	-0.36886	-0.34799	0.454562	0.574488
42	1	-1.29466	-0.19152	-0.36372	0.19482	-0.13183
43	8	0.484525	-0.52136	-0.61169	0.030867	-0.06619
44	2	-0.1047	0.664928	0.503343	-0.15131	0.222903
45	5	0.275286	-0.4461	-0.62836	0.293535	0.422611
46	7	-1.06863	0.771032	-0.13556	-0.05265	-0.15341
47	10	0.23167	-0.39483	0.378084	-0.01674	-0.05436
48	10	0.046785	0.599216	-0.08031	-0.20669	0.18767
49	6	0.172692	0.61368	-0.00939	-0.03567	0.070279
50	6	0.337794	-0.38823	0.493163	0.1217	-0.22504
51	8	-0.00384	-0.32722	-0.4843	0.650652	-0.64075
52	0	-0.23863	0.688637	0.159441	-0.02287	0.367885

53	11	0.174643	-0.37931	0.148202	0.149029	0.107236
54	12	0.369431	-0.36222	-0.10193	0.532704	-0.00111
55	6	0.460604	0.511221	-0.03738	-0.35881	-0.07492
56	11	0.428497	0.566894	0.300943	-0.27028	0.064901
57	11	0.221119	-0.43318	0.287604	-0.28488	-0.02018
58	4	0.429467	0.632907	0.403308	0.209772	-0.17075
59	2	-0.01772	-0.34697	0.099972	0.179849	0.109085
60	4	0.314618	-0.49886	0.06095	-0.57978	-0.0471
61	8	0.447487	-0.49909	-0.51014	0.056886	-0.16508
62	4	0.479009	0.532916	0.143784	-0.25616	0.033122
63	13	-0.2082	-0.39863	0.110611	-0.35391	0.217417
64	4	0.510406	0.504206	0.089037	-0.4792	-0.28122
65	13	0.072784	0.652389	0.228934	-0.00754	0.098953
66	3	0.515228	0.52658	-0.06786	-0.24859	-0.03425
67	4	0.378022	-0.3543	0.631394	0.229037	-0.1489
68	9	0.079178	-0.43491	-0.22581	-0.14197	0.067259
69	6	0.43555	-0.41472	-0.18276	0.369641	-0.11026
70	13	0.307345	-0.49864	0.112861	-0.48951	-0.03451
71	2	0.099326	-0.43436	-0.01656	-0.16131	0.130375
72	2	0.619017	0.500245	0.153352	-0.40097	-0.15016
73	4	0.261215	-0.399	0.545679	-0.13994	-0.07242
74	9	0.420581	0.634775	0.240972	0.193651	-0.01138
75	13	0.044186	0.558131	-0.07237	-0.42054	0.154459
76	9	-0.41291	-0.33036	-0.09738	0.027616	0.317912
77	3	0.114251	-0.45985	-0.13403	-0.37049	-0.55821
78	2	0.149082	-0.38803	0.230113	-0.02255	-0.00443
79	13	-0.12791	-0.42984	0.142717	-0.48621	0.14651
80	1	-0.87763	0.764416	0.076287	0.014638	-0.36693
81	8	0.347725	-0.39723	-0.35673	0.469246	-0.34281
82	9	-0.27963	-0.30388	0.072852	0.222388	0.241783
83	9	0.187446	0.55789	0.27587	-0.45073	0.045476
84	3	0.244012	-0.39007	0.203124	0.064748	-0.40126
85	10	-0.05832	0.598124	-0.16713	-0.25198	0.287035
86	9	0.343678	0.640593	0.173923	0.252207	-0.02349
87	3	-0.53178	0.648765	0.24916	-0.54433	-0.14396
88	12	0.418114	-0.43876	-0.32308	0.248481	-0.0784
89	9	-0.08109	-0.2912	0.536306	0.336041	0.033092
90	6	0.14391	-0.31283	0.373934	0.421317	0.050763
91	13	0.014701	-0.45613	0.207174	-0.53767	0.094613
92	12	0.508023	0.610559	-0.20592	0.397478	-0.15081
93	0	-0.35717	0.748595	0.160763	0.424655	0.082178
94	12	0.391258	-0.42018	-0.27113	0.306223	-0.22541
95	6	0.33457	-0.38257	0.41042	0.195274	0.030004

96	7	-1.10065	0.738401	-0.29283	-0.25766	0.040973
97	4	0.470013	0.627129	0.559982	0.088578	-0.08486
98	9	0.197938	0.548307	-0.26877	-0.30945	0.159235
99	3	0.494559	-0.44128	0.365255	-0.15743	-0.34495
100	1	-1.26741	-0.14138	0.209059	0.228851	-0.15517
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
700	7	-0.9879	0.795137	-0.33482	0.300566	-0.27967

## 6. Reduksi menggunakan PCA dengan 6 variabel fitur baru.

no	playlist	pca1	pca2	pca3	pca4	pca5	pca6
1	5	0.231265	-0.48271	-0.52019	0.012135	0.473969	-0.43233
2	11	0.019342	-0.37475	0.575705	-0.24897	0.080159	-0.00111
3	6	0.083597	0.684155	0.202254	0.376243	-0.03749	0.183279
4	3	0.327128	-0.48119	-0.18497	-0.30859	-0.03803	0.154428
5	2	0.155986	0.66781	0.239623	0.224091	0.066383	0.043486
6	6	0.213198	-0.37235	0.352747	0.106779	-0.03699	0.199307
7	13	-0.22343	-0.29291	0.545976	0.120568	0.273098	0.083161
8	9	0.300049	-0.47058	-0.39231	-0.03513	-0.11167	0.194091
9	0	-0.67088	-0.19968	0.393497	0.473913	0.338146	0.370183
10	12	0.410438	-0.51446	-0.4221	-0.22267	0.059703	-0.15935
11	3	-0.25352	-0.41067	-0.2665	-0.3842	-0.44227	-0.16109
12	5	-0.10885	-0.35925	-0.16653	0.33659	0.66451	-0.35449
13	12	0.441466	0.548259	-0.2964	-0.06949	0.194046	-0.10617
14	5	0.430253	-0.51853	-0.59208	0.056193	0.260659	-0.52283
15	5	0.090228	-0.35081	-0.15948	0.525331	0.571935	-0.29604
16	0	-0.133	0.572266	-0.2688	-0.50954	0.373133	0.291237
17	3	0.240056	-0.38394	0.094423	0.162781	-0.0135	0.158673
18	7	-0.88528	0.742997	-0.33648	0.086093	-0.23618	0.110275
19	7	-1.1949	-0.27278	-0.33955	-0.26624	-0.10043	-0.0027
20	11	0.348535	-0.38023	0.460184	0.131019	-0.107	0.059159
21	7	-1.32714	-0.16145	0.076057	0.115466	0.01146	0.015851
22	13	0.179812	-0.45495	0.192115	-0.44353	0.033678	0.099355
23	1	-1.05089	-0.22557	0.199849	-0.10297	-0.25414	-0.24242
24	11	0.332005	0.577097	-0.02851	-0.15625	0.075497	0.124682
25	7	-0.81025	-0.26785	0.265505	-0.15692	-0.31253	-0.32303
26	4	0.397105	-0.3818	0.529941	0.072798	-0.16578	-0.04058
27	4	0.335913	-0.51065	-0.00078	-0.54069	-0.21645	-0.06353
28	6	0.23205	-0.41634	0.117852	-0.05667	-0.02571	0.163056
29	13	0.157281	-0.36342	0.511434	0.005603	0.049549	-0.01066

30	0	-0.1676	0.728406	0.289961	0.317466	0.232327	0.31767
31	9	0.28378	-0.39756	0.301136	0.026411	-0.04405	0.107293
32	10	0.1715	-0.37293	0.215676	0.20215	-0.0982	0.32126
33	13	0.301444	-0.40455	0.388238	-0.01701	0.087048	-0.09855
34	10	0.348377	0.560023	0.285323	-0.37339	0.036937	0.071654
35	6	0.46293	0.627604	-0.0993	0.452122	0.046175	0.085607
36	6	0.271035	-0.38152	0.045862	0.320106	-0.14454	0.25542
37	5	0.386602	-0.51347	-0.49744	-0.01154	0.335967	-0.5094
38	4	0.467721	0.614639	0.570252	-0.03316	-0.10372	-0.03479
39	6	0.451506	0.620517	-0.34683	0.665153	0.265388	-0.17651
40	1	-1.44525	-0.1909	-0.13045	-0.07316	-0.01291	0.063164
41	5	-0.007	-0.36886	-0.34799	0.454562	0.574488	-0.357
42	1	-1.29466	-0.19152	-0.36372	0.19482	-0.13183	0.110847
43	8	0.484525	-0.52136	-0.61169	0.030867	-0.06619	-0.12187
44	2	-0.1047	0.664928	0.503343	-0.15131	0.222903	0.011714
45	5	0.275286	-0.4461	-0.62836	0.293535	0.422611	-0.33781
46	7	-1.06863	0.771032	-0.13556	-0.05265	-0.15341	0.00334
47	10	0.23167	-0.39483	0.378084	-0.01674	-0.05436	0.08119
48	10	0.046785	0.599216	-0.08031	-0.20669	0.18767	0.218925
49	6	0.172692	0.61368	-0.00939	-0.03567	0.070279	0.041104
50	6	0.337794	-0.38823	0.493163	0.1217	-0.22504	0.141031
51	8	-0.00384	-0.32722	-0.4843	0.650652	-0.64075	-0.04206
52	0	-0.23863	0.688637	0.159441	-0.02287	0.367885	0.179414
53	11	0.174643	-0.37931	0.148202	0.149029	0.107236	0.154362
54	12	0.369431	-0.36222	-0.10193	0.532704	-0.00111	-0.0357
55	6	0.460604	0.511221	-0.03738	-0.35881	-0.07492	-0.01916
56	11	0.428497	0.566894	0.300943	-0.27028	0.064901	-0.03372
57	11	0.221119	-0.43318	0.287604	-0.28488	-0.02018	-0.05476
58	4	0.429467	0.632907	0.403308	0.209772	-0.17075	-0.10848
59	2	-0.01772	-0.34697	0.099972	0.179849	0.109085	0.384032
60	4	0.314618	-0.49886	0.06095	-0.57978	-0.0471	-0.07078
61	8	0.447487	-0.49909	-0.51014	0.056886	-0.16508	0.023697
62	4	0.479009	0.532916	0.143784	-0.25616	0.033122	-0.29034
63	13	-0.2082	-0.39863	0.110611	-0.35391	0.217417	0.177584
64	4	0.510406	0.504206	0.089037	-0.4792	-0.28122	-0.18474
65	13	0.072784	0.652389	0.228934	-0.00754	0.098953	0.228045
66	3	0.515228	0.52658	-0.06786	-0.24859	-0.03425	-7.7E-05
67	4	0.378022	-0.3543	0.631394	0.229037	-0.1489	-0.15753
68	9	0.079178	-0.43491	-0.22581	-0.14197	0.067259	0.297432
69	6	0.43555	-0.41472	-0.18276	0.369641	-0.11026	0.087284
70	13	0.307345	-0.49864	0.112861	-0.48951	-0.03451	-0.06226
71	2	0.099326	-0.43436	-0.01656	-0.16131	0.130375	0.068815
72	2	0.619017	0.500245	0.153352	-0.40097	-0.15016	-0.11572

73	4	0.261215	-0.399	0.545679	-0.13994	-0.07242	-0.01787
74	9	0.420581	0.634775	0.240972	0.193651	-0.01138	0.13748
75	13	0.044186	0.558131	-0.07237	-0.42054	0.154459	0.276739
76	9	-0.41291	-0.33036	-0.09738	0.027616	0.317912	0.317897
77	3	0.114251	-0.45985	-0.13403	-0.37049	-0.55821	-0.48134
78	2	0.149082	-0.38803	0.230113	-0.02255	-0.00443	0.039901
79	13	-0.12791	-0.42984	0.142717	-0.48621	0.14651	0.086065
80	1	-0.87763	0.764416	0.076287	0.014638	-0.36693	-0.17856
81	8	0.347725	-0.39723	-0.35673	0.469246	-0.34281	0.194231
82	9	-0.27963	-0.30388	0.072852	0.222388	0.241783	0.32515
83	9	0.187446	0.55789	0.27587	-0.45073	0.045476	-0.06281
84	3	0.244012	-0.39007	0.203124	0.064748	-0.40126	-0.347
85	10	-0.05832	0.598124	-0.16713	-0.25198	0.287035	0.347124
86	9	0.343678	0.640593	0.173923	0.252207	-0.02349	0.170472
87	3	-0.53178	0.648765	0.24916	-0.54433	-0.14396	-0.25067
88	12	0.418114	-0.43876	-0.32308	0.248481	-0.0784	-0.00834
89	9	-0.08109	-0.2912	0.536306	0.336041	0.033092	0.129461
90	6	0.14391	-0.31283	0.373934	0.421317	0.050763	0.28409
91	13	0.014701	-0.45613	0.207174	-0.53767	0.094613	-0.00802
92	12	0.508023	0.610559	-0.20592	0.397478	-0.15081	0.185932
93	0	-0.35717	0.748595	0.160763	0.424655	0.082178	0.253497
94	12	0.391258	-0.42018	-0.27113	0.306223	-0.22541	0.223921
95	6	0.33457	-0.38257	0.41042	0.195274	0.030004	-0.07301
96	7	-1.10065	0.738401	-0.29283	-0.25766	0.040973	-0.14436
97	4	0.470013	0.627129	0.559982	0.088578	-0.08486	-0.04196
98	9	0.197938	0.548307	-0.26877	-0.30945	0.159235	0.243567
99	3	0.494559	-0.44128	0.365255	-0.15743	-0.34495	-0.10491
100	1	-1.26741	-0.14138	0.209059	0.228851	-0.15517	-0.07344
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
700	7	-0.9879	0.795137	-0.33482	0.300566	-0.27967	0.032981

## 7. Reduksi menggunakan PCA dengan 7 variabel fitur baru.

no	playlist	pca1	pca2	pca3	pca4	pca5	pca6	pca7
1	5	0.231265	-0.48271	-0.52019	0.012135	0.473969	-0.43233	0.077344
2	11	0.019342	-0.37475	0.575705	-0.24897	0.080159	-0.00111	0.034268
3	6	0.083597	0.684155	0.202254	0.376243	-0.03749	0.183279	-0.46053
4	3	0.327128	-0.48119	-0.18497	-0.30859	-0.03803	0.154428	0.237051
5	2	0.155986	0.66781	0.239623	0.224091	0.066383	0.043486	-0.0591
6	6	0.213198	-0.37235	0.352747	0.106779	-0.03699	0.199307	0.030679

7	13	-0.22343	-0.29291	0.545976	0.120568	0.273098	0.083161	0.039676
8	9	0.300049	-0.47058	-0.39231	-0.03513	-0.11167	0.194091	-0.09592
9	0	-0.67088	-0.19968	0.393497	0.473913	0.338146	0.370183	-0.3233
10	12	0.410438	-0.51446	-0.4221	-0.22267	0.059703	-0.15935	0.094355
11	3	-0.25352	-0.41067	-0.2665	-0.3842	-0.44227	-0.16109	0.437826
12	5	-0.10885	-0.35925	-0.16653	0.33659	0.66451	-0.35449	0.122754
13	12	0.441466	0.548259	-0.2964	-0.06949	0.194046	-0.10617	0.365024
14	5	0.430253	-0.51853	-0.59208	0.056193	0.260659	-0.52283	-0.14711
15	5	0.090228	-0.35081	-0.15948	0.525331	0.571935	-0.29604	0.319644
16	0	-0.133	0.572266	-0.2688	-0.50954	0.373133	0.291237	0.253022
17	3	0.240056	-0.38394	0.094423	0.162781	-0.0135	0.158673	0.231313
18	7	-0.88528	0.742997	-0.33648	0.086093	-0.23618	0.110275	-0.35005
19	7	-1.1949	-0.27278	-0.33955	-0.26624	-0.10043	-0.0027	0.031364
20	11	0.348535	-0.38023	0.460184	0.131019	-0.107	0.059159	0.036549
21	7	-1.32714	-0.16145	0.076057	0.115466	0.01146	0.015851	0.16537
22	13	0.179812	-0.45495	0.192115	-0.44353	0.033678	0.099355	0.153123
23	1	-1.05089	-0.22557	0.199849	-0.10297	-0.25414	-0.24242	-0.02029
24	11	0.332005	0.577097	-0.02851	-0.15625	0.075497	0.124682	0.231736
25	7	-0.81025	-0.26785	0.265505	-0.15692	-0.31253	-0.32303	-0.12977
26	4	0.397105	-0.3818	0.529941	0.072798	-0.16578	-0.04058	0.04098
27	4	0.335913	-0.51065	-0.00078	-0.54069	-0.21645	-0.06353	-0.02234
28	6	0.23205	-0.41634	0.117852	-0.05667	-0.02571	0.163056	0.024402
29	13	0.157281	-0.36342	0.511434	0.005603	0.049549	-0.01066	0.138019
30	0	-0.1676	0.728406	0.289961	0.317466	0.232327	0.31767	-0.11312
31	9	0.28378	-0.39756	0.301136	0.026411	-0.04405	0.107293	0.115008
32	10	0.1715	-0.37293	0.215676	0.20215	-0.0982	0.32126	-0.08043
33	13	0.301444	-0.40455	0.388238	-0.01701	0.087048	-0.09855	0.129569
34	10	0.348377	0.560023	0.285323	-0.37339	0.036937	0.071654	0.146809
35	6	0.46293	0.627604	-0.0993	0.452122	0.046175	0.085607	0.117922
36	6	0.271035	-0.38152	0.045862	0.320106	-0.14454	0.25542	-0.15715
37	5	0.386602	-0.51347	-0.49744	-0.01154	0.335967	-0.5094	-0.05202
38	4	0.467721	0.614639	0.570252	-0.03316	-0.10372	-0.03479	0.010177
39	6	0.451506	0.620517	-0.34683	0.665153	0.265388	-0.17651	0.012159
40	1	-1.44525	-0.1909	-0.13045	-0.07316	-0.01291	0.063164	0.17708
41	5	-0.007	-0.36886	-0.34799	0.454562	0.574488	-0.357	0.001325
42	1	-1.29466	-0.19152	-0.36372	0.19482	-0.13183	0.110847	-0.05636
43	8	0.484525	-0.52136	-0.61169	0.030867	-0.06619	-0.12187	-0.2467
44	2	-0.1047	0.664928	0.503343	-0.15131	0.222903	0.011714	-0.27916
45	5	0.275286	-0.4461	-0.62836	0.293535	0.422611	-0.33781	0.18659
46	7	-1.06863	0.771032	-0.13556	-0.05265	-0.15341	0.00334	-0.12621
47	10	0.23167	-0.39483	0.378084	-0.01674	-0.05436	0.08119	-0.05248
48	10	0.046785	0.599216	-0.08031	-0.20669	0.18767	0.218925	0.002597
49	6	0.172692	0.61368	-0.00939	-0.03567	0.070279	0.041104	-0.13122

50	6	0.337794	-0.38823	0.493163	0.1217	-0.22504	0.141031	-0.25999
51	8	-0.00384	-0.32722	-0.4843	0.650652	-0.64075	-0.04206	0.044851
52	0	-0.23863	0.688637	0.159441	-0.02287	0.367885	0.179414	-0.01755
53	11	0.174643	-0.37931	0.148202	0.149029	0.107236	0.154362	0.181945
54	12	0.369431	-0.36222	-0.10193	0.532704	-0.00111	-0.0357	0.15038
55	6	0.460604	0.511221	-0.03738	-0.35881	-0.07492	-0.01916	-0.16478
56	11	0.428497	0.566894	0.300943	-0.27028	0.064901	-0.03372	0.198246
57	11	0.221119	-0.43318	0.287604	-0.28488	-0.02018	-0.05476	0.005565
58	4	0.429467	0.632907	0.403308	0.209772	-0.17075	-0.10848	-0.3432
59	2	-0.01772	-0.34697	0.099972	0.179849	0.109085	0.384032	0.230012
60	4	0.314618	-0.49886	0.06095	-0.57978	-0.0471	-0.07078	0.025201
61	8	0.447487	-0.49909	-0.51014	0.056886	-0.16508	0.023697	-0.24064
62	4	0.479009	0.532916	0.143784	-0.25616	0.033122	-0.29034	-0.09987
63	13	-0.2082	-0.39863	0.110611	-0.35391	0.217417	0.177584	-0.07617
64	4	0.510406	0.504206	0.089037	-0.4792	-0.28122	-0.18474	0.004322
65	13	0.072784	0.652389	0.228934	-0.00754	0.098953	0.228045	0.341543
66	3	0.515228	0.52658	-0.06786	-0.24859	-0.03425	-7.7E-05	0.090202
67	4	0.378022	-0.3543	0.631394	0.229037	-0.1489	-0.15753	0.005659
68	9	0.079178	-0.43491	-0.22581	-0.14197	0.067259	0.297432	0.082193
69	6	0.43555	-0.41472	-0.18276	0.369641	-0.11026	0.087284	-0.05237
70	13	0.307345	-0.49864	0.112861	-0.48951	-0.03451	-0.06226	-0.02028
71	2	0.099326	-0.43436	-0.01656	-0.16131	0.130375	0.068815	0.00425
72	2	0.619017	0.500245	0.153352	-0.40097	-0.15016	-0.11572	-0.05031
73	4	0.261215	-0.399	0.545679	-0.13994	-0.07242	-0.01787	-0.04382
74	9	0.420581	0.634775	0.240972	0.193651	-0.01138	0.13748	0.259527
75	13	0.044186	0.558131	-0.07237	-0.42054	0.154459	0.276739	-0.07212
76	9	-0.41291	-0.33036	-0.09738	0.027616	0.317912	0.317897	-0.11749
77	3	0.114251	-0.45985	-0.13403	-0.37049	-0.55821	-0.48134	0.231248
78	2	0.149082	-0.38803	0.230113	-0.02255	-0.00443	0.039901	-0.01642
79	13	-0.12791	-0.42984	0.142717	-0.48621	0.14651	0.086065	-0.18763
80	1	-0.87763	0.764416	0.076287	0.014638	-0.36693	-0.17856	-0.31658
81	8	0.347725	-0.39723	-0.35673	0.469246	-0.34281	0.194231	-0.09539
82	9	-0.27963	-0.30388	0.072852	0.222388	0.241783	0.32515	-0.00562
83	9	0.187446	0.55789	0.27587	-0.45073	0.045476	-0.06281	-0.31676
84	3	0.244012	-0.39007	0.203124	0.064748	-0.40126	-0.347	0.174708
85	10	-0.05832	0.598124	-0.16713	-0.25198	0.287035	0.347124	0.137095
86	9	0.343678	0.640593	0.173923	0.252207	-0.02349	0.170472	0.043405
87	3	-0.53178	0.648765	0.24916	-0.54433	-0.14396	-0.25067	-0.25978
88	12	0.418114	-0.43876	-0.32308	0.248481	-0.0784	-0.00834	-0.00824
89	9	-0.08109	-0.2912	0.536306	0.336041	0.033092	0.129461	-0.34458
90	6	0.14391	-0.31283	0.373934	0.421317	0.050763	0.28409	0.231143
91	13	0.014701	-0.45613	0.207174	-0.53767	0.094613	-0.00802	-0.07689
92	12	0.508023	0.610559	-0.20592	0.397478	-0.15081	0.185932	0.047981

93	0	-0.35717	0.748595	0.160763	0.424655	0.082178	0.253497	-0.5654
94	12	0.391258	-0.42018	-0.27113	0.306223	-0.22541	0.223921	-0.0608
95	6	0.33457	-0.38257	0.41042	0.195274	0.030004	-0.07301	0.025337
96	7	-1.10065	0.738401	-0.29283	-0.25766	0.040973	-0.14436	0.071852
97	4	0.470013	0.627129	0.559982	0.088578	-0.08486	-0.04196	-0.05191
98	9	0.197938	0.548307	-0.26877	-0.30945	0.159235	0.243567	0.201183
99	3	0.494559	-0.44128	0.365255	-0.15743	-0.34495	-0.10491	0.052938
100	1	-1.26741	-0.14138	0.209059	0.228851	-0.15517	-0.07344	0.096187
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
700	7	-0.9879	0.795137	-0.33482	0.300566	-0.27967	0.032981	-0.28166

## 8. Reduksi menggunakan PCA dengan 8 variabel fitur baru.

no	playlist	pca1	pca2	pca3	pca4	pca5	pca6	pca7	pca8
1	5	0.231265	-0.48271	-0.52019	0.012135	0.473969	-0.43233	0.077344	-0.14957
2	11	0.019342	-0.37475	-0.575705	-0.24897	0.080159	-0.001118	0.034268	0.060193
3	6	0.083597	0.684155	0.202254	0.376243	-0.03749	0.183279	-0.46053	0.08306
4	3	0.327128	-0.48119	0.18497	0.30859	0.03803	-0.154428	0.237051	0.006743
5	2	0.155986	0.66781	0.239623	0.224091	0.066383	0.043486	-0.0591	0.22473
6	6	0.213198	-0.37235	-0.352747	0.106779	-0.03699	0.199307	0.030679	-0.26744
7	13	-0.22343	0.29291	-0.545976	0.120568	0.273098	0.083161	0.039676	0.105058
8	9	0.300049	-0.47058	-0.39231	0.03513	0.11167	0.194091	-0.09592	0.00304
9	0	-0.67088	0.19968	-0.393497	0.473913	0.338146	0.370183	-0.32334	0.01063
10	12	0.410438	-0.51446	-0.4221	0.22267	0.059703	-0.15935	0.094355	0.145204
11	3	-0.25352	0.41067	-0.2665	-0.3842	-0.44227	0.16109	0.437826	-0.15042
12	5	-0.10885	0.35925	-0.16653	0.33659	0.66451	-0.35449	0.122754	0.11421
13	12	0.441466	0.548259	-0.2964	-0.06949	0.194046	-0.10617	0.365024	0.233071
14	5	0.430253	-0.51853	-0.59208	0.056193	0.260659	-0.52283	0.14711	-0.0714
15	5	0.090228	-0.35081	-0.15948	0.525331	0.571935	-0.29604	0.319644	-0.26872
16	0	-0.133	0.572266	-0.2688	-0.50954	0.373133	0.291237	0.253022	0.06882
17	3	0.240056	-0.38394	-0.094423	0.162781	-0.0135	0.158673	0.231313	-0.07787
18	7	-0.88528	-0.742997	-0.33648	0.086093	-0.23618	0.110275	0.35005	-0.191
19	7	-1.1949	-0.27278	-0.33955	-0.26624	0.10043	-0.0027	0.031364	-0.07851

20	11	0.348535	- 0.38023	0.460184	0.131019	-0.107	0.059159	0.036549	- 0.08506
21	7	- 1.32714	- 0.16145	0.076057	0.115466	0.01146	0.015851	0.16537	0.078758
22	13	0.179812	- 0.45495	0.192115	- 0.44353	0.033678	0.099355	0.153123	-0.1107
23	1	- 1.05089	- 0.22557	0.199849	- 0.10297	- 0.25414	- 0.24242	- 0.02029	0.127771
24	11	0.332005	0.577097	- 0.02851	- 0.15625	0.075497	0.124682	0.231736	0.068148
25	7	- 0.81025	- 0.26785	0.265505	- 0.15692	- 0.31253	- 0.32303	- 0.12977	0.03434
26	4	0.397105	-0.3818	0.529941	0.072798	- 0.16578	- 0.04058	0.04098	0.168314
27	4	0.335913	- 0.51065	0.00078	0.54069	0.21645	0.06353	0.02234	0.099276
28	6	0.23205	- 0.41634	0.117852	- 0.05667	- 0.02571	0.163056	0.024402	0.24597
29	13	0.157281	- 0.36342	0.511434	0.005603	0.049549	- 0.01066	0.138019	0.13358
30	0	-0.1676	0.728406	0.289961	0.317466	0.232327	0.31767	- 0.11312	0.09344
31	9	0.28378	- 0.39756	0.301136	0.026411	- 0.04405	0.107293	0.115008	- 0.06766
32	10	0.1715	- 0.37293	0.215676	0.20215	-0.0982	0.32126	- 0.08043	-0.3636
33	13	0.301444	- 0.40455	0.388238	- 0.01701	0.087048	- 0.09855	0.129569	- 0.12912
34	10	0.348377	0.560023	0.285323	- 0.37339	0.036937	0.071654	0.146809	- 0.09369
35	6	0.46293	0.627604	-0.0993	0.452122	0.046175	0.085607	0.117922	- 0.15656
36	6	0.271035	- 0.38152	0.045862	0.320106	- 0.14454	0.25542	- 0.15715	0.22824
37	5	0.386602	- 0.51347	0.49744	0.01154	0.335967	-0.5094	- 0.05202	0.18443
38	4	0.467721	0.614639	0.570252	- 0.03316	0.10372	0.03479	0.010177	0.035964
39	6	0.451506	0.620517	- 0.34683	0.665153	0.265388	- 0.17651	0.012159	- 0.35759
40	1	- 1.44525	-0.1909	0.13045	0.07316	- 0.01291	0.063164	0.17708	-0.1023
41	5	-0.007	- 0.36886	0.34799	0.454562	0.574488	-0.357	0.001325	- 0.03541
42	1	- 1.29466	0.19152	0.36372	0.19482	- 0.13183	0.110847	- 0.05636	0.04596
43	8	0.484525	- 0.52136	0.61169	0.030867	- 0.06619	- 0.12187	-0.2467	- 0.02078
44	2	-0.1047	0.664928	0.503343	- 0.15131	0.222903	0.011714	- 0.27916	0.144446
45	5	0.275286	-0.4461	- 0.62836	0.293535	0.422611	- 0.33781	0.18659	- 0.01776
46	7	- 1.06863	0.771032	- 0.13556	- 0.05265	- 0.15341	0.00334	- 0.12621	0.21353
47	10	0.23167	- 0.39483	0.378084	- 0.01674	- 0.05436	0.08119	- 0.05248	0.08076
48	10	0.046785	0.599216	- 0.08031	- 0.20669	0.18767	0.218925	0.002597	0.043265
49	6	0.172692	0.61368	- 0.00939	- 0.03567	0.070279	0.041104	- 0.13122	0.382133
50	6	0.337794	- 0.38823	0.493163	0.1217	- 0.22504	0.141031	- 0.25999	0.41448

51	8	-	0.00384	0.32722	-0.4843	0.65065 2	-	0.64075	0.04206	0.04485 1	-	0.02707
52	0	-	0.23863	0.68863 7	0.15944 1	-	0.02287	0.36788 5	0.17941 4	-	0.01755 8	0.20551
53	11	0.17464 3	-	0.37931	0.14820 2	0.14902 9	0.10723 6	0.15436 2	0.18194 5	-	0.28721	
54	12	0.36943 1	-	0.36222	0.10193	0.53270 4	-	0.00111	-0.0357	0.15038	0.23034 6	
55	6	0.46060 4	0.51122 1	-	0.03738	0.35881	-	0.07492	0.01916	0.16478	0.03306	
56	11	0.42849 7	0.56689 4	0.30094 3	-	0.27028	0.06490 1	-	0.03372	0.19824 6	-	0.09535
57	11	0.22111 9	-	0.43318	0.28760 4	0.28488	0.02018	-	0.05476	0.00556 5	0.19183 3	
58	4	0.42946 7	0.63290 7	0.40330 8	0.20977 2	-	0.17075	0.10848	-	-0.3432	0.31979 2	
59	2	-	0.01772	0.34697	0.09997 2	0.17984 9	0.10908 5	0.38403	0.23001 2	-	0.20805	
60	4	0.31461 8	-	0.49886	0.06095	0.57978	-	-0.0471	0.07078	0.02520 1	0.08801 8	
61	8	0.44748 7	-	0.49909	0.51014	0.05688 6	0.16508	-	0.02369 7	-	0.24064	0.03658
62	4	0.47900 9	0.53291 6	0.14378 4	-	0.25616	0.03312 2	-	0.29034	0.09987	0.13947 4	
63	13	-0.2082	-	0.39863	0.11061 1	0.35391	-	0.21741 7	0.17758 4	-	0.01189 6	
64	4	0.51040 6	0.50420 6	0.08903 7	-0.4792	-	0.28122	0.18474	-	0.00432 2	0.05884 7	
65	13	0.07278 4	0.65238 9	0.22893 4	-	0.00754	0.09895 3	0.22804 5	0.34154 3	-	0.29479	
66	3	0.51522 8	0.52658	-	0.06786	0.24859	0.03425	-	-7.7E- 05	0.09020	0.11091 2	5
67	4	0.37802 2	-0.3543	0.63139 4	0.22903 7	-	-0.1489	-	0.00565 0.15753	0.02225 9	0.02225 8	
68	9	0.07917 8	-	0.43491	0.22581	0.14197	-	0.06725 9	0.29743 2	0.08219 3	-	0.10264
69	6	0.43555	-	0.41472	0.18276	-	0.36964 1	-	0.08728 4	-	-	0.13908
70	13	0.30734 5	-	0.49864	0.11286 1	-	0.48951	0.03451	0.06226	0.02028	-	0.11026
71	2	0.09932 6	-	0.43436	0.01656	0.16131	-	0.13037 5	0.06881 5	0.00425	-	0.18462
72	2	0.61901 7	0.50024 5	0.15335 2	-	0.40097	0.15016	-	0.11572	-	0.08914 5	
73	4	0.26121 5	-0.399	0.54567 9	-	0.13994	0.07242	-	0.01787	-	0.04382 3	0.03577
74	9	0.42058 1	0.63477 5	0.24097 2	0.19365 1	-	0.01138	0.13748	-	0.25952 7	0.09968 7	
75	13	0.04418 6	0.55813 1	-	0.07237	0.42054	-	0.15445 9	0.27673 9	-	0.07212	0.27136
76	9	-	0.41291	0.33036	-	0.09738	0.02761 6	0.31791 2	0.31789 7	-	0.11749 3	0.15468
77	3	0.11425 1	-	0.45985	0.13403	0.37049	-	0.55821	0.48134	-	0.23124 8	0.05134
78	2	0.14908 2	-	0.38803	0.23011 3	0.02255	-	0.00443	0.03990 1	-	0.01642 4	0.26530
79	13	-	0.12791	0.42984	0.14271 7	-	0.48621	0.14651	0.08606 5	-	0.18763 3	0.04787
80	1	-	0.87763	0.76441 6	0.07628 7	0.01463 8	-	0.36693	0.17856	-	0.31658	-0.0771
81	8	0.34772 5	-	0.39723	0.35673	0.46924 6	-	0.34281	0.19423 1	-	0.09539	0.08992 5

82	9	-	0.27963	0.30388	0.07285 2	0.22238 8	0.24178 3	0.32515	- 0.00562	0.09193
83	9	0.18744 6	0.55789	0.27587	- 0.45073	0.04547 6	-	0.06281	- 0.31676	0.07562 5
84	3	0.24401 2	- 0.39007	0.20312 4	0.06474 8	- 0.40126	-0.347	0.17470 8	-	0.02169
85	10	- 0.05832	0.59812 4	- 0.16713	- 0.25198	0.28703 5	0.34712 4	0.13709 5	-	0.10636
86	9	0.34367 8	0.64059 3	0.17392 3	0.25220 7	- 0.02349	0.17047 2	0.04340 5	0.04049 5	
87	3	- 0.53178	0.64876 5	0.24916	- 0.54433	0.14396	0.25067	0.25978	-	0.13361
88	12	0.41811 4	- 0.43876	0.32308	0.24848 1	-0.0784	- 0.00834	0.00824	- 0.15537	7
89	9	- 0.08109	-0.2912	0.53630 6	0.33604 1	0.03309 2	0.12946 1	- 0.34458	0.01896 4	
90	6	0.14391	- 0.31283	0.37393 4	0.42131 7	0.05076 3	0.28409	0.23114 3	- 0.35525	
91	13	0.01470 1	- 0.45613	0.20717 4	- 0.53767	0.09461 3	0.00802	0.07689	0.08265	
92	12	0.50802 3	0.61055 9	- 0.20592	0.39747 8	- 0.15081	0.18593 2	0.04798 1	0.16264	
93	0	- 0.35717	0.74859 5	0.16076 3	0.42465 5	0.08217 8	0.25349 7	-0.5654	-	0.00849
94	12	0.39125 8	- 0.42018	0.27113	0.30622 3	- 0.22541	0.22392 1	-0.0608	0.13880 3	
95	6	0.33457	- 0.38257	0.41042	0.19527 4	0.03000 4	- 0.07301	0.02533 7	-	0.32964
96	7	- 1.10065	0.73840 1	- 0.29283	- 0.25766	0.04097 3	- 0.14436	0.07185 2	-	0.01543
97	4	0.47001 3	0.62712 9	0.55998 2	0.08857 8	- 0.08486	- 0.04196	- 0.05191	-	0.07756
98	9	0.19793 8	0.54830 7	- 0.26877	- 0.30945	0.15923 5	0.24356 7	0.20118 3	-	0.02825
99	3	0.49455 9	- 0.44128	0.36525 5	- 0.15743	- 0.34495	- 0.10491	0.05293 8	0.04348 8	
100	1	- 1.26741	- 0.14138	- 0.20905 9	0.22885 1	- 0.15517	- 0.07344	0.09618 7	0.11034	
-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	
-	-	-	-	-	-	-	-	-	-	
700	7	-0.9879	0.79513 7	- 0.33482	0.30056 6	- 0.27967	0.03298 1	- 0.28166	-	0.07605

## 9. Reduksi menggunakan PCA dengan 9 variabel fitur baru.

no	pla ylis t	pca1	pca2	pca3	pca4	pca5	pca6	pca7	pca8	pca9
1	5	0.2312 65	- 0.4827 1	- 0.5201 9	0.0121 35	0.4739 69	- 0.4323 3	0.0773 44	- 0.1495 7	- 0.0638 3
2	11	0.0193 42	- 0.3747 5	0.5757 05	- 0.2489 7	0.0801 59	- 0.0011 1	0.0342 68	0.0601 93	0.0586 02
3	6	0.0835 97	0.6841 55	0.2022 54	0.3762 43	- 0.0374 9	0.1832 79	- 0.4605 3	- 0.0830 6	0.1129 9

4	3	0.3271 28	0.4811 9	- 1849 7	0.3085 9	- 0.0380 3	0.1544 28	0.2370 51	0.0067 43	0.0447 73
5	2	0.1559 86	0.6678 1	0.2396 23	0.2240 91	0.0663 83	0.0434 86	- 0.0591	0.2247 3	0.0353 13
6	6	0.2131 98	- 0.3723 5	0.3527 47	0.1067 79	- 0.0369 9	0.1993 07	0.0306 79	- 0.2674 4	0.2731 67
7	13	- 0.2234 3	- 0.2929 1	0.5459 76	0.1205 68	0.2730 98	0.0831 61	0.0396 76	0.1050 58	0.0118 7
8	9	0.3000 49	- 0.4705 8	0.3923 1	0.0351 3	- 0.1116 7	0.1940 91	- 0.0959 2	- 0.0030 4	- 0.0647 6
9	0	- 0.6708 8	- 0.1996 8	0.3934 97	0.4739 13	0.3381 46	0.3701 83	- 0.3233	0.0106 34	0.0846 07
10	12	0.4104 38	- 0.5144 6	- 0.4221	- 0.2226 7	0.0597 03	- 0.1593 5	0.0943 55	0.1452 04	- 0.0551 9
11	3	0.2535 2	- 0.4106 7	- 0.2665	- 0.3842	- 0.4422 7	- 0.1610 9	0.4378 26	- 0.1504 2	0.1841 83
12	5	0.1088 5	- 0.3592 5	- 0.1665 3	0.3365 9	0.6645 1	- 0.3544 9	0.1227 54	- 0.1142 1	0.0715 83
13	12	0.4414 66	0.5482 59	- 0.2964	- 0.0694 9	0.1940 46	- 0.1061 7	0.3650 24	0.2330 71	0.0118 99
14	5	0.4302 53	- 0.5185 3	- 0.5920 8	0.0561 93	0.2606 59	- 0.5228 3	0.1471 1	- 0.0714	- 0.0207 5
15	5	0.0902 28	- 0.3508 1	- 0.1594 8	0.5253 31	0.5719 35	- 0.2960 4	0.3196 44	- 0.2687 2	- 0.2262 2
16	0	-0.133	0.5722 66	- 0.2688	- 0.5095 4	0.3731 33	0.2912 37	0.2530 22	- 0.0688 2	- 0.1431 6
17	3	0.2400 56	- 0.3839 4	0.0944 23	0.1627 81	- 0.0135	0.1586 73	0.2313 13	- 0.0778 7	0.1981 23
18	7	0.8852 8	0.7429 97	- 0.3364 8	0.0860 93	- 0.2361 8	0.1102 75	- 0.3500 5	-0.191	- 0.0136 5
19	7	- 1.1949	- 0.2727 8	0.3395 5	- 0.2662 4	- 0.1004 3	- 0.0027	0.0313 64	- 0.0785 1	0.0279 39
20	11	0.3485 35	- 0.3802 3	0.4601 84	0.1310 19	-0.107	0.0591 59	0.0365 49	- 0.0850 6	- 0.0483 6
21	7	- 1.3271 4	- 0.1614 5	0.0760 57	0.1154 66	0.0114 6	0.0158 51	0.1653 7	0.0787 58	- 0.0328 4
22	13	0.1798 12	- 0.4549 5	0.1921 15	- 0.4435 3	0.0336 78	0.0993 55	0.1531 23	- 0.1107	- 0.1160 5
23	1	- 1.0508 9	- 0.2255 7	0.1998 49	- 0.1029 7	- 0.2541 4	- 0.2424 2	- 0.0202 9	0.1277 71	0.1281 86
24	11	0.3320 05	0.5770 97	- 0.0285 1	- 0.1562 5	0.0754 97	0.1246 82	0.2317 36	0.0681 48	0.0086 25

25	7	- 0.8102 5	- 0.2678 5	0.2655 05	- 0.1569 2	- 0.3125 3	- 0.3230 3	- 0.1297 7	- 0.0343 4	- 0.1737
26	4	0.3971 05	- 0.3818	0.5299 41	0.0727 98	- 0.1657 8	- 0.0405 8	0.0409 8	0.1683 14	- 0.0556
27	4	0.3359 13	- 0.5106 5	- 0.0007 8	0.5406 9	- 0.2164 5	- 0.0635 3	- 0.0223 4	0.0992 76	- 0.0280 5
28	6	0.2320 5	- 0.4163 4	0.1178 52	- 0.0566 7	- 0.0257 1	0.1630 56	0.0244 02	- 0.2459 7	0.0839 22
29	13	0.1572 81	- 0.3634 2	0.5114 34	0.0056 03	0.0495 49	- 0.0106 6	0.1380 19	0.1335 8	- 0.0890 5
30	0	- 0.1676	0.7284 06	0.2899 61	0.3174 66	0.2323 27	0.3176 7	- 0.1131 2	- 0.0934 4	-8.6E- 05
31	9	0.2837 8	- 0.3975 6	0.3011 36	0.0264 11	- 0.0440 5	0.1072 93	0.1150 08	- 0.0676 6	- 0.0112 7
32	10	0.1715	- 0.3729 3	0.2156 76	0.2021 5	- 0.0982	0.3212 6	- 0.0804 3	- 0.3636	- 0.1256
33	13	0.3014 44	- 0.4045 5	0.3882 38	- 0.0170 1	0.0870 48	- 0.0985 5	0.1295 69	- 0.1291 2	- 0.0734 7
34	10	0.3483 77	0.5600 23	0.2853 23	- 0.3733 9	0.0369 37	0.0716 54	0.1468 09	- 0.0936 9	- 0.1562 3
35	6	0.4629 3	0.6276 04	- 0.0993	0.4521 22	0.0461 75	0.0856 07	0.1179 22	- 0.1565 6	0.2108 71
36	6	0.2710 35	- 0.3815 2	0.0458 62	0.3201 06	- 0.1445 4	0.2554 2	- 0.1571 5	- 0.2282 4	0.1954 25
37	5	0.3866 02	- 0.5134 7	0.4974 4	- 0.0115 4	0.3359 67	- 0.5094	- 0.0520 2	0.1844 3	- 0.0536 3
38	4	0.4677 21	0.6146 39	0.5702 52	- 0.0331 6	- 0.1037 2	- 0.0347 9	0.0101 77	0.0359 64	- 0.0905 6
39	6	0.4515 06	0.6205 17	- 0.3468 3	0.6651 53	0.2653 88	- 0.1765 1	0.0121 59	- 0.3575 9	0.1178 33
40	1	- 1.4452 5	- 0.1909	0.1304 5	- 0.0731 6	- 0.0129 1	0.0631 64	0.1770 8	- 0.1023	- 0.0797 1
41	5	-0.007	- 0.3688 6	- 0.3479 9	0.4545 62	0.5744 88	-0.357	0.0013 25	- 0.0354 1	- 0.0075 5
42	1	- 1.2946 6	0.1915 2	- 0.3637 2	0.1948 2	- 0.1318 3	0.1108 47	- 0.0563 6	0.0459 6	- 0.3745 5
43	8	0.4845 25	- 0.5213 6	- 0.6116 9	0.0308 67	- 0.0661 9	- 0.1218 7	- 0.2467	- 0.0207 8	0.0518 81
44	2	- 0.1047	0.6649 28	0.5033 43	- 0.1513 1	0.2229 03	0.0117 14	- 0.2791 6	0.1444 46	0.1681 21
45	5	0.2752 86	- 0.4461	- 0.6283 6	0.2935 35	0.4226 11	- 0.3378 1	0.1865 9	- 0.0177 6	- 0.2702 6

46	7	- 1.0686 3	0.7710 32	- 0.1355 6	- 0.0526 5	- 0.1534 1	0.0033 4	- 0.1262 1	- 0.2135 3	0.0709 37
47	10	0.2316 7	- 0.3948 3	0.3780 84	- 0.0167 4	- 0.0543 6	0.0811 9	- 0.0524 8	- 0.0807 6	0.0379 14
48	10	0.0467 85	0.5992 16	- 0.0803 1	- 0.2066 9	0.1876 7	0.2189 25	0.0025 97	0.0432 65	0.1845 21
49	6	0.1726 92	0.6136 8	- 0.0093 9	- 0.0356 7	0.0702 79	0.0411 04	- 0.1312 2	0.3821 33	0.2315 56
50	6	0.3377 94	- 0.3882 3	0.4931 63	0.1217	- 0.2250 4	0.1410 31	- 0.2599 9	- 0.4144 8	0.1574 69
51	8	- 0.0038 4	- 0.3272 2	- 0.4843	0.6506 52	- 0.6407 5	- 0.0420 6	0.0448 51	- 0.0270 7	0.0131 45
52	0	- 0.2386 3	0.6886 37	0.1594 41	- 0.0228 7	0.3678 85	0.1794 14	- 0.0175 5	0.2055 18	- 0.0542 4
53	11	0.1746 43	- 0.3793 1	0.1482 02	0.1490 29	0.1072 36	0.1543 62	0.1819 45	- 0.2872 1	- 0.0100 5
54	12	0.3694 31	- 0.3622 2	- 0.1019 3	0.5327 04	- 0.0011 1	- 0.0357	0.1503 8	0.2303 46	- 0.3361 5
55	6	0.4606 04	0.5112 21	- 0.0373 8	- 0.3588 1	- 0.0749 2	- 0.0191 6	- 0.1647 8	- 0.0330 6	0.1398 33
56	11	0.4284 97	0.5668 94	0.3009 43	- 0.2702 8	0.0649 01	- 0.0337 2	0.1982 46	- 0.0953 5	0.0732 54
57	11	0.2211 19	- 0.4331 8	0.2876 04	- 0.2848 8	- 0.0201 8	- 0.0547 6	0.0055 65	0.1918 33	- 0.0116 6
58	4	0.4294 67	0.6329 07	0.4033 08	0.2097 72	- 0.1707 5	- 0.1084 8	- 0.3432	0.3197 92	0.0825 86
59	2	- 0.0177 2	- 0.3469 7	0.0999 72	0.1798 49	0.1090 85	0.3840 32	0.2300 12	- 0.2080 5	- 0.1147
60	4	0.3146 18	- 0.4988 6	0.0609 5	- 0.5797 8	- 0.0471	- 0.0707 8	0.0252 01	0.0880 18	- 0.3689 5
61	8	0.4474 87	- 0.4990 9	0.5101 4	0.0568 86	- 0.1650 8	0.0236 97	- 0.2406 4	0.0365 8	0.0212 85
62	4	0.4790 09	0.5329 16	0.1437 84	- 0.2561 6	0.0331 22	- 0.2903 4	- 0.0998 7	0.1394 74	- 0.0077 9
63	13	- 0.2082	- 0.3986 3	0.1106 11	- 0.3539 1	0.2174 17	0.1775 84	- 0.0761 7	0.0118 96	0.0308 31
64	4	0.5104 06	0.5042 06	0.0890 37	- 0.4792	- 0.2812 2	- 0.1847 4	0.0043 22	0.0588 47	0.0189 37
65	13	0.0727 84	0.6523 89	0.2289 34	- 0.0075 4	0.0989 53	0.2280 45	0.3415 43	- 0.2947 9	0.0808 42
66	3	0.5152 28	0.5265 8	- 0.0678 6	- 0.2485 9	- 0.0342 5	-7.7E- 05	0.0902 02	0.1109 15	0.1314 95

67	4	0.3780 22	- 0.3543	0.6313 94	0.2290 37	- 0.1489	- 0.1575 3	0.0056 59	0.0222 58	- 0.1135 2
68	9	0.0791 78	- 0.4349 1	- 0.2258 1	- 0.1419 7	0.0672 59	0.2974 32	0.0821 93	- 0.1026 4	- 0.0987 4
69	6	0.4355 5	- 0.4147 2	- 0.1827 6	0.3696 41	- 0.1102 6	0.0872 84	- 0.0523 7	- 0.1390 8	0.1476 05
70	13	0.3073 45	- 0.4986 4	0.1128 61	- 0.4895 1	0.0345 1	- 0.0622 6	- 0.0202 8	- 0.1102 6	- 0.1369 8
71	2	0.0993 26	- 0.4343 6	- 0.0165 6	- 0.1613 1	0.1303 75	0.0688 15	0.0042 5	- 0.1846 2	- 0.1372 1
72	2	0.6190 17	0.5002 45	0.1533 52	- 0.4009 7	0.1501 6	- 0.1157 2	- 0.0503 1	0.0891 45	- 0.0048 5
73	4	0.2612 15	-0.399	0.5456 79	- 0.1399 4	0.0724 2	- 0.0178 7	- 0.0438 2	0.0357 73	- 0.0056 2
74	9	0.4205 81	0.6347 75	0.2409 72	0.1936 51	- 0.0113 8	0.1374 8	0.2595 27	0.0996 87	0.0116 04
75	13	0.0441 86	0.5581 31	- 0.0723 7	- 0.4205 4	0.1544 59	0.2767 39	- 0.0721 2	- 0.2713 6	0.1314 85
76	9	0.4129 1	0.3303 6	0.0973 8	0.0276 16	0.3179 12	0.3178 97	- 0.1174 9	0.1546 83	0.0745 85
77	3	0.1142 51	- 0.4598 5	0.1340 3	- 0.3704 9	- 0.5582 1	- 0.4813 4	0.2312 48	- 0.0513 4	0.1178 33
78	2	0.1490 82	- 0.3880 3	0.2301 13	- 0.0225 5	- 0.0044 3	0.0399 01	- 0.0164 2	0.2653 04	0.0603 59
79	13	- 0.1279 1	- 0.4298 4	0.1427 17	- 0.4862 1	0.1465 1	0.0860 65	- 0.1876 3	0.0478 73	0.0219 09
80	1	- 0.8776 3	0.7644 16	0.0762 87	0.0146 38	- 0.3669 3	- 0.1785 6	- 0.3165 8	- 0.0771	0.0244 09
81	8	0.3477 25	- 0.3972 3	0.3567 3	0.4692 46	- 0.3428 1	0.1942 31	- 0.0953 9	0.0899 25	- 0.0034 2
82	9	- 0.2796 3	- 0.3038 8	0.0728 52	0.2223 88	0.2417 83	0.3251 5	- 0.0056 2	0.0919 3	0.1563 85
83	9	0.1874 46	0.5578 9	0.2758 7	- 0.4507 3	0.0454 76	- 0.0628 1	- 0.3167 6	0.0756 25	0.0544 58
84	3	0.2440 12	- 0.3900 7	0.2031 24	0.0647 48	- 0.4012 6	- 0.347	0.1747 08	- 0.0216 9	0.0713 7
85	10	- 0.0583 2	0.5981 24	0.1671 3	- 0.2519 8	0.2870 35	0.3471 24	0.1370 95	- 0.1063 6	0.1670 48
86	9	0.3436 78	0.6405 93	0.1739 23	0.2522 07	- 0.0234 9	0.1704 72	0.0434 05	0.0404 95	- 0.0176 8
87	3	- 0.5317 8	0.6487 65	0.2491 6	- 0.5443 3	0.1439 6	- 0.2506 7	- 0.2597 8	- 0.1336 1	0.2041 28

88	12	0.4181 14	- 0.4387 6	- 0.3230 8	0.2484 81	- 0.0784	- 0.0083 4	- 0.0082 4	0.1553 77	- 0.1031 3
89	9	- 0.0810 9	- 0.2912	0.5363 06	0.3360 41	0.0330 92	0.1294 61	- 0.3445 8	0.0189 64	0.0194 1
90	6	0.1439 1	- 0.3128 3	0.3739 34	0.4213 17	0.0507 63	0.2840 9	0.2311 43	- 0.3552 5	0.2205 63
91	13	0.0147 01	- 0.4561 3	0.2071 74	- 0.5376 7	0.0946 13	- 0.0080 2	- 0.0768 9	- 0.0826 5	- 0.0564 8
92	12	0.5080 23	0.6105 59	- 0.2059 2	0.3974 78	- 0.1508 1	0.1859 32	0.0479 81	0.1626 4	0.0263 5
93	0	- 0.3571 7	0.7485 95	0.1607 63	0.4246 55	0.0821 78	0.2534 97	- 0.5654	- 0.0084 9	- 0.0392 9
94	12	0.3912 58	- 0.4201 8	- 0.2711 3	0.3062 23	- 0.2254 1	0.2239 21	- 0.0608	0.1388 03	0.0001 54
95	6	0.3345 7	- 0.3825 7	0.4104 2	0.1952 74	0.0300 04	- 0.0730 1	0.0253 37	- 0.3296 4	0.1404 99
96	7	1.1006 5	0.7384 01	- 0.2928 3	- 0.2576 6	0.0409 73	- 0.1443 6	0.0718 52	- 0.0154 3	- 0.0369
97	4	0.4700 13	0.6271 29	0.5599 82	0.0885 78	- 0.0848 6	- 0.0419 6	- 0.0519 1	- 0.0775 6	- 0.1290 9
98	9	0.1979 38	0.5483 07	- 0.2687 7	- 0.3094 5	0.1592 35	0.2435 67	0.2011 83	- 0.0282 5	0.0765 01
99	3	0.4945 59	- 0.4412 8	0.3652 55	- 0.1574 3	- 0.3449 5	- 0.1049 1	0.0529 38	0.0434 88	0.0741 07
100	1	- 1.2674 1	- 0.1413 8	0.2090 59	0.2288 51	- 0.1551 7	- 0.0734 4	0.0961 87	0.1103 4	- 0.1140 1
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-
700	7	- 0.9879	0.7951 37	- 0.3348 2	0.3005 66	- 0.2796 7	0.0329 81	- 0.2816 6	- 0.0760 5	0.0069 26

#### 10. Reduksi menggunakan PCA dengan 10 variabel fitur baru.

no	playlist	pca1	pca2	pca3	pca4	pca5	pca6	pca7	pca8	pca9	pca10
1	5	0.231 265	- 0.482 71	- 0.520 19	0.012 135	0.473 969	- 0.432 33	0.077 344	- 0.149 57	- 0.063 83	0.047 158
2	11	0.019 342	- 0.374 75	0.575 705	- 0.248 97	0.080 159	- 0.001 11	0.034 268	0.060 193	0.058 602	- 0.200 91

3	6	0.083 597	0.684 155	0.202 254	0.376 243	- 0.037 49	0.183 279	- 0.460 53	- 0.083 06	0.112 99	- 0.020 65
4	3	0.327 128	- 0.481 19	- 0.184 97	- 0.308 59	- 0.038 03	0.154 428	0.237 051	0.006 743	0.044 773	0.232 447
5	2	0.155 986	0.667 81	0.239 623	0.224 091	0.066 383	0.043 486	- 0.059 1	0.224 73	0.035 313	- 0.013 13
6	6	0.213 198	- 0.372 35	0.352 747	0.106 779	- 0.036 99	0.199 307	0.030 679	- 0.267 44	0.273 167	0.023 381
7	13	- 0.223 43	- 0.292 91	0.545 976	0.120 568	0.273 098	0.083 161	0.039 676	0.105 058	0.011 87	- 0.120 02
8	9	0.300 049	- 0.470 58	- 0.392 31	- 0.035 13	- 0.111 67	0.194 091	- 0.095 92	- 0.003 04	- 0.064 76	0.066 244
9	0	- 0.670 88	- 0.199 68	0.393 497	0.473 913	0.338 146	0.370 183	- 0.323 3	0.010 634	0.084 607	0.012 06
10	12	0.410 438	- 0.514 46	- 0.422 1	- 0.222 67	0.059 703	- 0.159 35	0.094 355	0.145 204	- 0.055 19	0.038 538
11	3	- 0.253 52	- 0.410 67	- 0.266 5	- 0.384 2	- 0.442 27	- 0.161 09	0.437 826	- 0.150 42	0.184 183	0.044 002
12	5	- 0.108 85	- 0.359 25	- 0.166 53	- 0.336 59	0.664 51	- 0.354 49	0.122 754	- 0.114 21	0.071 583	- 0.063 92
13	12	0.441 466	0.548 259	- 0.296 4	- 0.069 49	0.194 046	- 0.106 17	0.365 024	0.233 071	0.011 899	- 0.048 23
14	5	0.430 253	- 0.518 53	- 0.592 08	- 0.056 193	0.260 659	- 0.522 83	- 0.147 11	0.071 4	- 0.020 75	- 0.096 83
15	5	0.090 228	- 0.350 81	- 0.159 48	0.525 331	0.571 935	- 0.296 04	0.319 644	- 0.268 72	- 0.226 22	0.118 225
16	0	-0.133	0.572 266	- 0.268 8	- 0.509 54	0.373 133	0.291 237	0.253 022	- 0.068 82	- 0.143 16	0.182 779
17	3	0.240 056	- 0.383 94	0.094 423	0.162 781	- 0.013 5	0.158 673	0.231 313	- 0.077 87	0.198 123	0.163 69
18	7	- 0.885 28	0.742 997	- 0.336 48	0.086 093	- 0.236 18	0.110 275	- 0.350 05	-0.191 0.013 65	- 0.013 65	0.136 826
19	7	- 1.194 9	- 0.272 78	- 0.339 55	- 0.266 24	- 0.100 43	- 0.002 7	0.031 364	- 0.078 51	0.027 939	0.162 533
20	11	0.348 535	- 0.380 23	0.460 184	0.131 019	-0.107	0.059 159	0.036 549	- 0.085 06	- 0.048 36	0.044 614
21	7	- 1.327 14	- 0.161 45	0.076 057	0.115 466	0.011 46	0.015 851	0.165 37	0.078 758	- 0.032 84	- 0.133 09
22	13	0.179 812	- 0.454 95	0.192 115	- 0.443 53	0.033 678	0.099 355	0.153 123	- 0.110 7	- 0.116 05	0.085 968
23	1	- 1.050 89	- 0.225 57	0.199 849	- 0.102 97	0.254 14	- 0.242 42	0.020 29	0.127 771	0.128 186	0.004 089

24	11	0.332 005	0.577 097	- 0.028 51	0.156 25	0.075 497	0.124 682	0.231 736	0.068 148	0.008 625	0.023 616
25	7	- 0.810 25	- 0.267 85	0.265 505	0.156 92	- 0.312 53	- 0.323 03	- 0.129 77	- 0.034 34	0.173 7	0.215 803
26	4	0.397 105	- 0.381 8	0.529 941	0.072 798	- 0.165 78	- 0.040 58	0.040 98	0.168 314	- 0.055 6	0.101 159
27	4	0.335 913	- 0.510 65	0.000 78	0.540 69	- 0.216 45	- 0.063 53	- 0.022 34	0.099 276	- 0.028 05	- 0.025 8
28	6	0.232 05	- 0.416 34	0.117 852	0.056 67	- 0.025 71	0.163 056	0.024 402	- 0.245 97	0.083 922	0.065 911
29	13	0.157 281	- 0.363 42	0.511 434	0.005 603	0.049 549	- 0.010 66	0.138 019	0.133 58	- 0.089 05	- 0.219 95
30	0	- 0.167 6	0.728 406	0.289 961	0.317 466	0.232 327	0.317 67	- 0.113 12	- 0.093 44	-8.6E- 05	0.117 515
31	9	0.283 78	- 0.397 56	0.301 136	0.026 411	- 0.044 05	0.107 293	0.115 008	- 0.067 66	- 0.011 27	- 0.020 27
32	10	0.171 5	- 0.372 93	0.215 676	0.202 15	- 0.098 2	0.321 26	- 0.080 43	- 0.363 6	- 0.125 6	0.005 918
33	13	0.301 444	- 0.404 55	0.388 238	- 0.017 01	0.087 048	- 0.098 55	0.129 569	- 0.129 12	- 0.073 47	- 0.030 49
34	10	0.348 377	0.560 023	0.285 323	- 0.373 39	0.036 937	0.071 654	0.146 809	- 0.093 69	- 0.156 23	- 0.072 84
35	6	0.462 93	0.627 604	- 0.099 3	0.452 122	0.046 175	0.085 607	0.117 922	- 0.156 56	0.210 871	- 0.009 71
36	6	0.271 035	- 0.381 52	0.045 862	0.320 106	- 0.144 54	0.255 42	- 0.157 15	- 0.228 24	0.195 425	0.050 917
37	5	0.386 602	- 0.513 47	0.497 44	- 0.011 54	0.335 967	- 0.509 4	- 0.052 02	- 0.184 43	- 0.053 63	0.025 315
38	4	0.467 721	0.614 639	0.570 252	- 0.033 16	0.103 72	- 0.034 79	0.010 177	0.035 964	- 0.090 56	0.004 672
39	6	0.451 506	0.620 517	- 0.346 83	0.665 153	0.265 388	- 0.176 51	0.012 159	- 0.357 59	0.117 833	- 0.045 72
40	1	- 1.445 25	- 0.190 9	- 0.130 45	- 0.073 16	- 0.012 91	0.063 164	0.177 08	- 0.102 3	- 0.079 71	- 0.053 78
41	5	-0.007	0.368 86	- 0.347 99	0.454 562	0.574 488	-0.357	0.001 325	- 0.035 41	- 0.007 55	0.161 299
42	1	- 1.294 66	0.191 52	- 0.363 72	0.194 82	- 0.131 83	0.110 847	- 0.056 36	- 0.045 96	- 0.374 55	0.191 758
43	8	0.484 525	- 0.521 36	- 0.611 69	0.030 867	- 0.066 19	0.121 87	- 0.246 7	- 0.020 78	0.051 881	- 0.056 71
44	2	- 0.104 7	0.664 928	0.503 343	- 0.151 31	0.222 903	0.011 714	- 0.279 16	0.144 446	0.168 121	0.011 576

45	5	0.275 286	- 0.446 1	- 0.628 36	0.293 535	0.422 611	- 0.337 81	0.186 59	- 0.017 76	- 0.270 26	0.130 977
46	7	- 1.068 63	0.771 032	- 0.135 56	0.052 65	0.153 41	0.003 34	- 0.126 21	- 0.213 53	0.070 937	0.078 497
47	10	0.231 67	- 0.394 83	0.378 084	- 0.016 74	0.054 36	0.081 19	- 0.052 48	- 0.080 76	0.037 914	- 0.056 19
48	10	0.046 785	0.599 216	- 0.080 31	- 0.206 69	0.187 67	0.218 925	0.002 597	0.043 265	0.184 521	0.106 919
49	6	0.172 692	0.613 68	- 0.009 39	- 0.035 67	0.070 279	0.041 104	- 0.131 22	0.382 133	0.231 556	- 0.049 13
50	6	0.337 794	- 0.388 23	0.493 163	0.121 7	- 0.225 04	0.141 031	- 0.259 99	0.414 48	0.157 469	- 0.136 5
51	8	- 0.003 84	- 0.327 22	- 0.484 3	0.650 652	- 0.640 75	- 0.042 06	0.044 851	- 0.027 07	0.013 145	- 0.020 34
52	0	- 0.238 63	0.688 637	0.159 441	- 0.022 87	0.367 885	0.179 414	- 0.017 55	0.205 518	- 0.054 24	0.160 199
53	11	0.174 643	- 0.379 31	0.148 202	0.149 029	0.107 236	0.154 362	0.181 945	- 0.287 21	- 0.010 05	0.022 043
54	12	0.369 431	- 0.362 22	- 0.101 93	0.532 704	- 0.001 11	- 0.035 7	0.150 38	0.230 346	- 0.336 15	0.126 127
55	6	0.460 604	0.511 221	- 0.037 38	- 0.358 81	- 0.074 92	- 0.019 16	- 0.164 78	- 0.033 06	0.139 833	- 0.092 01
56	11	0.428 497	0.566 894	0.300 943	- 0.270 28	0.064 901	- 0.033 72	0.198 246	- 0.095 35	0.073 254	- 0.113 99
57	11	0.221 119	- 0.433 18	0.287 604	- 0.284 88	- 0.020 18	- 0.054 76	0.005 565	0.191 833	- 0.011 66	- 0.027 28
58	4	0.429 467	0.632 907	0.403 308	0.209 772	- 0.170 75	- 0.108 48	- 0.343 2	0.319 792	0.082 586	- 0.063 77
59	2	- 0.017 72	- 0.346 97	0.099 972	0.179 849	0.109 085	0.384 032	0.230 012	- 0.208 05	- 0.114 7	- 0.063 76
60	4	0.314 618	- 0.498 86	0.060 95	- 0.579 78	- 0.047 1	- 0.070 78	0.025 201	0.088 018	- 0.368 95	0.359 705
61	8	0.447 487	- 0.499 09	- 0.510 14	0.056 886	- 0.165 08	0.023 697	- 0.240 64	0.036 58	0.021 285	- 0.100 26
62	4	0.479 009	0.532 916	0.143 784	- 0.256 16	0.033 122	- 0.290 34	- 0.099 87	0.139 474	- 0.007 79	- 0.118 26
63	13	- 0.208 2	- 0.398 63	0.110 611	- 0.353 91	0.217 417	0.177 584	- 0.076 17	0.011 896	0.030 831	0.153 548
64	4	0.510 406	0.504 206	0.089 037	- 0.479 2	- 0.281 22	- 0.184 74	0.004 322	0.058 847	0.018 937	- 0.019 68
65	13	0.072 784	0.652 389	0.228 934	- 0.007 54	0.098 953	0.228 045	0.341 543	- 0.294 79	0.080 842	0.073 38

66	3	0.515 228	0.526 58	- 0.067 86	0.248 59	- 0.034 25	-7.7E- 05	0.090 202	0.110 915	0.131 495	0.134 252
67	4	0.378 022	0.354 3	- 0.631 394	0.229 037	- 0.148 9	- 0.157 53	0.005 659	0.022 258	- 0.113 52	0.143 642
68	9	0.079 178	0.434 91	- 0.225 81	0.141 97	0.067 259	0.297 432	0.082 193	0.102 64	- 0.098 74	0.062 727
69	6	0.435 55	- 0.414 72	- 0.182 76	0.369 641	- 0.110 26	0.087 284	- 0.052 37	- 0.139 08	0.147 605	0.072 3
70	13	0.307 345	- 0.498 64	0.112 861	0.489 51	- 0.034 51	0.062 26	0.020 28	0.110 26	- 0.136 98	- 0.066 43
71	2	0.099 326	- 0.434 36	- 0.016 56	0.161 31	0.130 375	0.068 815	0.004 25	- 0.184 62	- 0.137 21	- 0.025 14
72	2	0.619 017	0.500 245	0.153 352	0.400 97	- 0.150 16	- 0.115 72	- 0.050 31	0.089 145	- 0.004 85	- 0.088 09
73	4	0.261 215	-0.399	0.545 679	0.139 94	- 0.072 42	- 0.017 87	- 0.043 82	0.035 773	- 0.005 62	0.022 867
74	9	0.420 581	0.634 775	0.240 972	0.193 651	- 0.011 38	0.137 48	0.259 527	0.099 687	0.011 604	- 0.046 99
75	13	0.044 186	0.558 131	- 0.072 37	0.420 54	0.154 459	0.276 739	- 0.072 12	0.271 36	0.131 485	- 0.196 4
76	9	- 0.412 91	- 0.330 36	- 0.097 38	0.027 616	0.317 912	0.317 897	- 0.117 49	0.154 683	0.074 585	0.114 079
77	3	0.114 251	- 0.459 85	- 0.134 03	0.370 49	- 0.558 21	- 0.481 34	0.231 248	- 0.051 34	0.117 833	0.104 255
78	2	0.149 082	- 0.388 03	0.230 113	0.022 55	- 0.004 43	0.039 901	- 0.016 42	0.265 304	0.060 359	- 0.003 79
79	13	- 0.127 91	- 0.429 84	0.142 717	0.486 21	0.146 51	0.086 065	- 0.187 63	0.047 873	0.021 909	0.112 81
80	1	- 0.877 63	0.764 416	0.076 287	0.014 638	- 0.366 93	- 0.178 56	- 0.316 58	- 0.077 1	0.024 409	- 0.149 72
81	8	0.347 725	- 0.397 23	- 0.356 73	0.469 246	- 0.342 81	0.194 231	- 0.095 39	0.089 925	- 0.003 42	0.040 402
82	9	- 0.279 63	- 0.303 88	0.072 852	0.222 388	0.241 783	0.325 15	- 0.005 62	0.091 93	0.156 385	0.126 242
83	9	0.187 446	0.557 89	0.275 87	0.450 73	0.045 476	- 0.062 81	- 0.316 76	0.075 625	0.054 458	0.014 212
84	3	0.244 012	- 0.390 07	0.203 124	0.064 748	- 0.401 26	-0.347	0.174 708	- 0.021 69	0.071 37	0.136 853
85	10	- 0.058 32	0.598 124	- 0.167 13	0.251 98	0.287 035	0.347 124	0.137 095	- 0.106 36	0.167 048	0.042 409
86	9	0.343 678	0.640 593	0.173 923	0.252 207	- 0.023 49	0.170 472	0.043 405	0.040 495	- 0.017 68	- 0.020 54

87	3	- 0.531 78	0.648 765	0.249 16	- 0.544 33	- 0.143 96	- 0.250 67	- 0.259 78	- 0.133 61	0.204 128	0.082 746
88	12	0.418 114	- 0.438 76	- 0.323 08	0.248 481	- 0.078 4	- 0.008 34	- 0.008 24	0.155 377	- 0.103 13	0.037 805
89	9	- 0.081 09	- 0.291 2	0.536 306	0.336 041	0.033 092	0.129 461	- 0.344 58	0.018 964	0.019 41	0.107 21
90	6	0.143 91	- 0.312 83	0.373 934	0.421 317	0.050 763	0.284 09	0.231 143	- 0.355 25	0.220 563	0.006 993
91	13	0.014 701	- 0.456 13	0.207 174	- 0.537 67	0.094 613	- 0.008 02	- 0.076 89	0.082 65	- 0.056 48	- 0.044 96
92	12	0.508 023	0.610 559	- 0.205 92	0.397 478	- 0.150 81	0.185 932	0.047 981	0.162 64	0.026 35	- 0.107 25
93	0	- 0.357 17	0.748 595	0.160 763	0.424 655	0.082 178	0.253 497	- 0.565 4	0.008 49	- 0.039 29	0.094 552
94	12	0.391 258	- 0.420 18	- 0.271 13	0.306 223	- 0.225 41	0.223 921	- 0.060 8	0.138 803	0.000 154	- 0.008 5
95	6	0.334 57	- 0.382 57	0.410 42	0.195 274	0.030 004	- 0.073 01	0.025 337	- 0.329 64	0.140 499	- 0.177 43
96	7	- 1.100 65	0.738 401	- 0.292 83	- 0.257 66	0.040 973	- 0.144 36	0.071 852	- 0.015 43	- 0.036 9	- 0.049 02
97	4	0.470 013	0.627 129	0.559 982	0.088 578	- 0.084 86	- 0.041 96	- 0.051 91	- 0.077 56	- 0.129 09	0.100 704
98	9	0.197 938	0.548 307	- 0.268 77	- 0.309 45	0.159 235	0.243 567	0.201 183	- 0.028 25	0.076 501	0.049 789
99	3	0.494 559	- 0.441 28	0.365 255	- 0.157 43	- 0.344 95	- 0.104 91	0.052 938	0.043 488	0.074 107	0.077 311
100	1	- 1.267 41	- 0.141 38	0.209 059	0.228 851	- 0.155 17	- 0.073 44	0.096 187	0.110 34	- 0.114 01	- 0.071 35
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-
700	7	- 0.987 9	0.795 137	- 0.334 82	0.300 566	- 0.279 67	0.032 981	- 0.281 66	- 0.076 05	0.006 926	0.178 077

## 11. Reduksi menggunakan PCA dengan 11 variabel fitur baru.

no	pla ylis t	pca1	pca2	pca3	pca4	pca5	pca6	pca7	pca8	pca9	pca1 0	pca1 1
1	5	0.231 265	- 0.482 71	- 0.520 19	0.012 135	0.473 969	- 0.432 33	0.077 344	- 0.149 57	- 0.063 83	0.047 158	0.046 018

2	11	0.019 342	- 0.374 75	0.575 705	- 0.248 97	0.080 159	- 0.001 11	0.034 268	0.060 193	0.058 602	- 0.200 91	- 0.205 65
3	6	0.083 597	0.684 155	0.202 254	0.376 243	- 0.037 49	0.183 279	- 0.460 53	- 0.083 06	0.112 99	- 0.020 65	- 0.237 78
4	3	0.327 128	- 0.481 19	0.184 97	- 0.308 59	- 0.038 03	0.154 428	0.237 051	0.006 743	0.044 773	0.232 447	- 0.007 27
5	2	0.155 986	0.667 81	0.239 623	0.224 091	0.066 383	0.043 486	- 0.059 1	0.224 73	0.035 313	- 0.013 13	0.018 491
6	6	0.213 198	- 0.372 35	0.352 747	0.106 779	- 0.036 99	0.199 307	0.030 679	- 0.267 44	0.273 167	0.023 381	- 0.136 14
7	13	- 0.223 43	- 0.292 91	0.545 976	0.120 568	0.273 098	0.083 161	0.039 676	0.105 058	0.011 87	- 0.120 02	0.061 725
8	9	0.300 049	- 0.470 58	0.392 31	0.035 13	- 0.111 67	0.194 091	- 0.095 92	- 0.003 04	- 0.064 76	0.066 244	0.065 002
9	0	- 0.670 88	- 0.199 68	0.393 497	0.473 913	0.338 146	0.370 183	- 0.323 3	0.010 634	0.084 607	0.012 06	0.057 468
10	12	0.410 438	- 0.514 46	0.422 1	0.222 67	0.059 703	- 0.159 35	0.094 355	0.145 204	- 0.055 19	0.038 538	0.008 278
11	3	- 0.253 52	- 0.410 67	0.266 5	0.384 2	- 0.442 27	- 0.161 09	0.437 826	- 0.150 42	0.184 183	0.044 002	0.098 208
12	5	- 0.108 85	- 0.359 25	0.166 53	0.336 59	0.664 51	- 0.354 49	0.122 754	- 0.114 21	0.071 583	- 0.063 92	0.144 482
13	12	0.441 466	0.548 259	- 0.296 4	0.069 49	0.194 046	- 0.106 17	0.365 024	0.233 071	0.011 899	- 0.048 23	- 0.005 91
14	5	0.430 253	- 0.518 53	0.592 08	0.056 193	0.260 659	- 0.522 83	- 0.147 11	- 0.071 4	- 0.020 75	- 0.096 83	- 0.039 77
15	5	0.090 228	- 0.350 81	0.159 48	0.525 331	0.571 935	- 0.296 04	0.319 644	- 0.268 72	- 0.226 22	0.118 225	0.051 364
16	0	- 0.133	0.572 266	- 0.268 8	0.509 54	0.373 133	0.291 237	0.253 022	- 0.068 82	- 0.143 16	0.182 779	- 0.011 36
17	3	0.240 056	- 0.383 94	0.094 423	0.162 781	- 0.013 5	0.158 673	0.231 313	- 0.077 87	0.198 123	0.163 69	0.153 56
18	7	- 0.885 28	0.742 997	- 0.336 48	0.086 093	- 0.236 18	0.110 275	- 0.350 05	- 0.191	- 0.013 65	0.136 826	0.055 71
19	7	- 1.194 9	0.272 78	- 0.339 55	0.266 24	0.100 43	- 0.002 7	0.031 364	- 0.078 51	0.027 939	0.162 533	0.017 135
20	11	0.348 535	- 0.380 23	0.460 184	0.131 019	- 0.107	0.059 159	0.036 549	- 0.085 06	- 0.048 36	0.044 614	0.101 882
21	7	- 1.327 14	- 0.161 45	0.076 057	0.115 466	0.011 46	0.015 851	0.165 37	0.078 758	- 0.032 84	- 0.133 09	- 0.009 63
22	13	0.179 812	- 0.454 95	0.192 115	- 0.443 53	0.033 678	0.099 355	0.153 123	- 0.110 7	- 0.116 05	0.085 968	- 0.017 35

23	1	- 1.050 89	- 0.225 57	0.199 849	- 0.102 97	- 0.254 14	- 0.242 42	- 0.020 29	0.127 771	0.128 186	0.004 089	0.007 506
24	11	0.332 005	0.577 097	- 0.028 51	- 0.156 25	0.075 497	0.124 682	0.231 736	0.068 148	0.008 625	0.023 616	- 0.130 33
25	7	- 0.810 25	- 0.267 85	0.265 505	- 0.156 92	- 0.312 53	- 0.323 03	- 0.129 77	- 0.034 34	0.173 7	0.215 803	0.039 157
26	4	0.397 105	- 0.381 8	0.529 941	0.072 798	- 0.165 78	- 0.040 58	0.040 98	0.168 314	- 0.055 6	0.101 159	- 0.045 2
27	4	0.335 913	- 0.510 65	0.000 78	0.540 69	- 0.216 45	- 0.063 53	- 0.022 34	0.099 276	- 0.028 05	- 0.025 8	0.000 524
28	6	0.232 05	- 0.416 34	0.117 852	- 0.056 67	- 0.025 71	0.163 056	0.024 402	- 0.245 97	0.083 922	0.065 911	- 0.263 11
29	13	0.157 281	- 0.363 42	0.511 434	0.005 603	0.049 549	- 0.010 66	0.138 019	0.133 58	- 0.089 05	- 0.219 95	0.035 083
30	0	- 0.167 6	0.728 406	0.289 961	0.317 466	0.232 327	0.317 67	- 0.113 12	- 0.093 44	- 8.6E- 05	0.117 515	0.080 412
31	9	0.283 78	- 0.397 56	0.301 136	0.026 411	- 0.044 05	0.107 293	0.115 008	- 0.067 66	- 0.011 27	- 0.020 27	0.121 823
32	10	0.171 5	- 0.372 93	0.215 676	0.202 15	- 0.098 2	0.321 26	- 0.080 43	- 0.363 6	- 0.125 6	0.005 918	0.137 543
33	13	0.301 444	- 0.404 55	0.388 238	- 0.017 01	0.087 048	- 0.098 55	0.129 569	- 0.129 12	- 0.073 47	- 0.030 49	0.099 497
34	10	0.348 377	0.560 023	0.285 323	- 0.373 39	0.036 937	0.071 654	0.146 809	- 0.093 69	- 0.156 23	- 0.072 84	0.065 547
35	6	0.462 93	0.627 604	- 0.099 3	0.452 122	0.046 175	0.085 607	0.117 922	- 0.156 56	0.210 871	- 0.009 71	- 0.202 1
36	6	0.271 035	- 0.381 52	0.045 862	0.320 106	- 0.144 54	0.255 42	- 0.157 15	- 0.228 24	0.195 425	0.050 917	- 0.086 81
37	5	0.386 602	- 0.513 47	0.497 44	0.011 54	0.335 967	- 0.509 4	- 0.052 02	- 0.184 43	- 0.053 63	0.025 315	0.083 39
38	4	0.467 721	0.614 639	0.570 252	- 0.033 16	- 0.103 72	- 0.034 79	0.010 177	0.035 964	- 0.090 56	0.004 672	0.006 717
39	6	0.451 506	0.620 517	- 0.346 83	0.665 153	0.265 388	- 0.176 51	0.012 159	- 0.357 59	0.117 833	- 0.045 72	- 0.230 42
40	1	- 1.445 25	- 0.190 9	0.130 45	0.073 16	- 0.012 91	0.063 164	0.177 08	- 0.102 3	- 0.079 71	- 0.053 78	- 0.041 92
41	5	- 0.007	- 0.368 86	- 0.347 99	0.454 562	0.574 488	- 0.357	0.001 325	- 0.035 41	- 0.007 55	0.161 299	0.043 743
42	1	- 1.294 66	- 0.191 52	- 0.363 72	0.194 82	- 0.131 83	0.110 847	- 0.056 36	- 0.045 96	- 0.374 55	0.191 758	- 0.195 49
43	8	0.484 525	- 0.521 36	- 0.611 69	0.030 867	- 0.066 19	- 0.121 87	- 0.246 7	- 0.020 78	0.051 881	- 0.056 71	0.021 079

44	2	- 0.104 7	0.664 928	0.503 343	- 0.151 31	0.222 903	0.011 714	- 0.279 16	0.144 446	0.168 121	0.011 576	0.018 41
45	5	0.275 286	- 0.446 1	- 0.628 36	0.293 535	0.422 611	- 0.337 81	0.186 59	- 0.017 76	- 0.270 26	0.130 977	- 0.061 41
46	7	- 1.068 63	0.771 032	- 0.135 56	- 0.052 65	- 0.153 41	0.003 34	- 0.126 21	- 0.213 53	0.070 937	0.078 497	0.085 899
47	10	0.231 67	- 0.394 83	0.378 084	- 0.016 74	- 0.054 36	0.081 19	- 0.052 48	- 0.080 76	0.037 914	- 0.056 19	0.049 816
48	10	0.046 785	0.599 216	- 0.080 31	- 0.206 69	0.187 67	0.218 925	0.002 597	0.043 265	0.184 521	0.106 919	- 0.064 11
49	6	0.172 692	0.613 68	- 0.009 39	- 0.035 67	0.070 279	0.041 104	- 0.131 22	0.382 133	0.231 556	- 0.049 13	- 0.296 77
50	6	0.337 794	- 0.388 23	0.493 163	0.121 7	- 0.225 04	0.141 031	- 0.259 99	0.414 48	0.157 469	- 0.136 5	- 0.299 38
51	8	- 0.003 84	- 0.327 22	- 0.484 3	0.650 652	- 0.640 75	- 0.042 06	0.044 851	- 0.027 07	0.013 145	- 0.020 34	- 0.007 58
52	0	- 0.238 63	0.688 637	0.159 441	- 0.022 87	0.367 885	0.179 414	- 0.017 55	0.205 518	- 0.054 24	0.160 199	- 0.058 03
53	11	0.174 643	- 0.379 31	0.148 202	0.149 029	0.107 236	0.154 362	0.181 945	- 0.287 21	- 0.010 05	0.022 043	0.014 701
54	12	0.369 431	- 0.362 22	0.101 93	0.532 704	- 0.001 11	- 0.035 7	0.150 38	0.230 346	- 0.336 15	0.126 127	- 0.188 25
55	6	0.460 604	0.511 221	- 0.037 38	0.358 81	- 0.074 92	- 0.019 16	- 0.164 78	- 0.033 06	0.139 833	- 0.092 01	- 0.193 32
56	11	0.428 497	0.566 894	0.300 943	- 0.270 28	0.064 901	- 0.033 72	0.198 246	- 0.095 35	0.073 254	- 0.113 99	- 0.183 06
57	11	0.221 119	- 0.433 18	0.287 604	- 0.284 88	0.020 18	- 0.054 76	0.005 565	0.191 833	- 0.011 66	- 0.027 28	- 0.002 79
58	4	0.429 467	0.632 907	0.403 308	0.209 772	0.170 75	- 0.108 48	- 0.343 2	0.319 792	0.082 586	- 0.063 77	- 0.044 39
59	2	- 0.017 72	- 0.346 97	0.099 972	0.179 849	0.109 085	0.384 032	0.230 012	- 0.208 05	- 0.114 7	- 0.063 76	0.144 915
60	4	0.314 618	- 0.498 86	0.060 95	- 0.579 78	- 0.047 1	- 0.070 78	0.025 201	0.088 018	- 0.368 95	0.359 705	- 0.236 8
61	8	0.447 487	- 0.499 09	0.510 14	0.056 886	- 0.165 08	0.023 697	- 0.240 64	0.036 58	0.021 285	- 0.100 26	0.050 734
62	4	0.479 009	0.532 916	0.143 784	- 0.256 16	0.033 122	0.290 34	- 0.099 87	0.139 474	- 0.007 79	- 0.118 26	0.038 048
63	13	- 0.208 2	- 0.398 63	0.110 611	- 0.353 91	0.217 417	0.177 584	- 0.076 17	0.011 896	0.030 831	0.153 548	0.027 822
64	4	0.510 406	0.504 206	0.089 037	- 0.479 2	- 0.281 22	- 0.184 74	0.004 322	0.058 847	0.018 937	- 0.019 68	0.026 881

65	13	0.072 784	0.652 389	0.228 934	- 0.007 54	0.098 953	0.228 045	0.341 543	- 0.294 79	0.080 842	0.073 38	0.178 589
66	3	0.515 228	0.526 58	- 0.067 86	- 0.248 59	- 0.034 25	- 7.7E- 05	0.090 202	0.110 915	0.131 495	0.134 252	0.093 516
67	4	0.378 022	- 0.354 3	0.631 394	0.229 037	0.148 9	- 0.157 53	0.005 659	0.022 258	- 0.113 52	0.143 642	- 0.083 06
68	9	0.079 178	- 0.434 91	0.225 81	- 0.141 97	0.067 259	0.297 432	0.082 193	- 0.102 64	- 0.098 74	0.062 727	0.092 156
69	6	0.435 55	- 0.414 72	0.182 76	0.369 641	- 0.110 26	0.087 284	- 0.052 37	- 0.139 08	0.147 605	0.072 3	- 0.184 6
70	13	0.307 345	- 0.498 64	0.112 861	0.489 51	- 0.034 51	- 0.062 26	0.020 28	- 0.110 26	- 0.136 98	- 0.066 43	0.008 299
71	2	0.099 326	- 0.434 36	0.016 56	0.161 31	0.130 375	0.068 815	0.004 25	- 0.184 62	- 0.137 21	- 0.025 14	0.091 891
72	2	0.619 017	0.500 245	0.153 352	0.400 97	- 0.150 16	- 0.115 72	- 0.050 31	0.089 145	- 0.004 85	- 0.088 09	0.085 082
73	4	0.261 215	- 0.399	0.545 679	- 0.139 94	- 0.072 42	- 0.017 87	- 0.043 82	0.035 773	- 0.005 62	0.022 867	0.028 432
74	9	0.420 581	0.634 775	0.240 972	0.193 651	- 0.011 38	0.137 48	0.259 527	0.099 687	0.011 604	- 0.046 99	0.104 745
75	13	0.044 186	0.558 131	- 0.072 37	0.420 54	0.154 459	0.276 739	- 0.072 12	- 0.271 36	0.131 485	- 0.196 4	- 0.111 12
76	9	- 0.412 91	- 0.330 36	- 0.097 38	0.027 616	0.317 912	0.317 897	- 0.117 49	0.154 683	0.074 585	0.114 079	0.028 701
77	3	0.114 251	- 0.459 85	0.134 03	0.370 49	- 0.558 21	- 0.481 34	0.231 248	- 0.051 34	0.117 833	0.104 255	0.049 129
78	2	0.149 082	- 0.388 03	0.230 113	0.022 55	- 0.004 43	0.039 901	- 0.016 42	0.265 304	0.060 359	- 0.003 79	0.025 65
79	13	- 0.127 91	- 0.429 84	0.142 717	0.486 21	0.146 51	0.086 065	- 0.187 63	0.047 873	0.021 909	0.112 81	0.014 545
80	1	- 0.877 63	0.764 416	0.076 287	0.014 638	- 0.366 93	- 0.178 56	- 0.316 58	- 0.077 1	0.024 409	- 0.149 72	0.087 021
81	8	0.347 725	- 0.397 23	0.356 73	0.469 246	- 0.342 81	0.194 231	- 0.095 39	0.089 925	- 0.003 42	0.040 402	0.001 918
82	9	- 0.279 63	- 0.303 88	0.072 852	0.222 388	0.241 783	0.325 15	- 0.005 62	0.091 93	0.156 385	0.126 242	0.009 397
83	9	0.187 446	0.557 89	0.275 87	0.450 73	0.045 476	- 0.062 81	- 0.316 76	0.075 625	0.054 458	0.014 212	0.044 025
84	3	0.244 012	- 0.390 07	0.203 124	0.064 748	- 0.401 26	- 0.347	0.174 708	- 0.021 69	0.071 37	0.136 853	0.054 887
85	10	- 0.058 32	0.598 124	- 0.167 13	0.251 98	0.287 035	0.347 124	0.137 095	- 0.106 36	0.167 048	0.042 409	0.168 261

86	9	0.343 678	0.640 593	0.173 923	0.252 207	- 0.023 49	0.170 472	0.043 405	0.040 495	- 0.017 68	- 0.020 54	0.113 995
87	3	- 0.531 78	0.648 765	0.249 16	0.544 33	- 0.143 96	0.250 67	0.259 78	0.133 61	0.204 128	0.082 746	0.067 558
88	12	0.418 114	- 0.438 76	0.323 08	0.248 481	- 0.078 4	0.008 34	0.008 24	0.155 377	- 0.103 13	0.037 805	- 0.025 32
89	9	- 0.081 09	- 0.291 2	0.536 306	0.336 041	0.033 092	0.129 461	- 0.344 58	0.018 964	0.019 41	0.107 21	0.026 354
90	6	0.143 91	- 0.312 83	0.373 934	0.421 317	0.050 763	0.284 09	0.231 143	- 0.355 25	0.220 563	0.006 993	- 0.081 69
91	13	0.014 701	- 0.456 13	0.207 174	0.537 67	0.094 613	0.008 02	0.076 89	- 0.082 65	- 0.056 48	- 0.044 96	0.029 241
92	12	0.508 023	0.610 559	- 0.205 92	0.397 478	- 0.150 81	0.185 932	0.047 981	0.162 64	0.026 35	- 0.107 25	- 0.124 28
93	0	- 0.357 17	0.748 595	0.160 763	0.424 655	0.082 178	0.253 497	- 0.565 4	- 0.008 49	- 0.039 29	0.094 552	0.014 696
94	12	0.391 258	- 0.420 18	0.271 13	0.306 223	- 0.225 41	0.223 921	- 0.060 8	0.138 803	0.000 154	- 0.008 5	0.052 761
95	6	0.334 57	- 0.382 57	0.410 42	0.195 274	0.030 004	- 0.073 01	0.025 337	- 0.329 64	0.140 499	- 0.177 43	- 0.207 67
96	7	- 1.100 65	0.738 401	- 0.292 83	0.257 66	0.040 973	- 0.144 36	0.071 852	- 0.015 43	- 0.036 9	- 0.049 02	- 0.003 57
97	4	0.470 013	0.627 129	0.559 982	0.088 578	- 0.084 86	- 0.041 96	- 0.051 91	- 0.077 56	- 0.129 09	0.100 704	- 0.001 26
98	9	0.197 938	0.548 307	- 0.268 77	0.309 45	0.159 235	0.243 567	0.201 183	- 0.028 25	0.076 501	0.049 789	0.117 781
99	3	0.494 559	- 0.441 28	0.365 255	0.157 43	- 0.344 95	- 0.104 91	0.052 938	0.043 488	0.074 107	0.077 311	- 0.183 69
100	1	- 1.267 41	- 0.141 38	0.209 059	0.228 851	- 0.155 17	- 0.073 44	0.096 187	0.110 34	- 0.114 01	- 0.071 35	- 0.086 79
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-
700	7	- 0.987 9	0.795 137	- 0.334 82	0.300 566	- 0.279 67	0.032 981	- 0.281 66	- 0.076 05	0.006 926	0.178 077	0.032 35

### Lampiran C.2. Source Code PCA-KNN Pengujian Kedua

No	Source Code
1	public function KlasifikasiPCA(\$pca,\$nilaiK){
2	\$hasilShuffle['nilaiK'] = \$nilaiK;

No	<i>Source Code</i>
3	\$hasilShuffle['indexPca'] = \$pca; \$hasilShuffle['data'] = \$this->Data->ViewDataPca ("data_pca".\$pca); for (\$f=0; \$f<10; \$f++) {
4	\$data0 = array(); \$data1 = array(); \$data2 = array(); \$data3 = array(); \$data4 = array(); \$data5 = array(); \$data6 = array(); \$data7 = array(); \$data8 = array(); \$data9 = array(); \$data10 = array(); \$data11 = array(); \$data12 = array(); \$data13 = array(); \$allDataTesting = array(); // \$dataIndex0 = 0; \$dataIndex1 = 0; \$dataIndex2 = 0; \$dataIndex3 = 0; \$dataIndex4 = 0; \$dataIndex5 = 0; \$dataIndex6 = 0; \$dataIndex7 = 0; \$dataIndex8 = 0; \$dataIndex9 = 0; \$dataIndex10 = 0; \$dataIndex11 = 0; \$dataIndex12 = 0; \$dataIndex13 = 0; // \$dataHasil0 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); \$dataHasil1 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); \$dataHasil2 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); \$dataHasil3 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); \$dataHasil4 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0);

No	Source Code
42	\$dataHasil5 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
43	\$dataHasil6 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
44	\$dataHasil7 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
45	\$dataHasil8 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
46	\$dataHasil9 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
47	\$dataHasil10 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
48	\$dataHasil11 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
49	\$dataHasil12 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
50	\$dataHasil13 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
51	for(\$j=0;\$j<count(\$hasilShuffle['data']);\$j++) {
52	if(\$hasilShuffle['data'][\$j]->playlist == 0) {
53	if(count(\$data0)<5&& \$dataIndex0 == \$f*5) {
54	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
55	array_push(\$data0, \$hasilShuffle['data'][\$j]);
56	} else {
57	\$dataIndex0++;
58	}
59	}
60	if(\$hasilShuffle['data'][\$j]->playlist == 1) {
61	if(count(\$data1)<5&& \$dataIndex1 == \$f*5) {
62	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
63	array_push(\$data1, \$hasilShuffle['data'][\$j]);
64	} else {
65	\$dataIndex1++;
66	}
67	}
68	if(\$hasilShuffle['data'][\$j]->playlist == 2) {
69	if(count(\$data2)<5&& \$dataIndex2 == \$f*5) {
70	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]);
71	array_push(\$data2, \$hasilShuffle['data'][\$j]);
72	} else {
73	\$dataIndex2++;
74	}
75	}
76	if(\$hasilShuffle['data'][\$j]->playlist == 3) {
77	if(count(\$data3)<5&& \$dataIndex3 == \$f*5) {

No	Source Code
78	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data3, \$hasilShuffle['data'][\$j]); } else { \$dataIndex3++; } } if(\$hasilShuffle['data'][\$j]->playlist == 4){ if(count(\$data4)<5&& \$dataIndex4 == \$f*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data4, \$hasilShuffle['data'][\$j]); } else { \$dataIndex4++; } } if(\$hasilShuffle['data'][\$j]->playlist == 5){ if(count(\$data5)<5&& \$dataIndex5 == \$f*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data5, \$hasilShuffle['data'][\$j]); } else { \$dataIndex5++; } } if(\$hasilShuffle['data'][\$j]->playlist == 6){ if(count(\$data6)<5&& \$dataIndex6 == \$f*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data6, \$hasilShuffle['data'][\$j]); } else { \$dataIndex6++; } } if(\$hasilShuffle['data'][\$j]->playlist == 7){ if(count(\$data7)<5&& \$dataIndex7 == \$f*5){ array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data7, \$hasilShuffle['data'][\$j]); } else { \$dataIndex7++; } }

No	Source Code
114	}
115	}
116	if(\$hasilShuffle['data'][\$j]->playlist == 8) {
117	if(count(\$data8)<5&& \$dataIndex8 == \$f*5) {
118	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
119	array_push(\$data8, \$hasilShuffle['data'][\$j] );
120	}else{
121	\$dataIndex8++;
122	}
123	}
124	if(\$hasilShuffle['data'][\$j]->playlist == 9) {
125	if(count(\$data9)<5&& \$dataIndex9 == \$f*5) {
126	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
127	array_push(\$data9, \$hasilShuffle['data'][\$j] );
128	}else{
129	\$dataIndex9++;
130	}
131	}
132	if(\$hasilShuffle['data'][\$j]->playlist == 10) {
133	if(count(\$data10)<5&& \$dataIndex10 == \$f*5) {
134	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
135	array_push(\$data10, \$hasilShuffle['data'][\$j] );
136	}else{
137	\$dataIndex10++;
138	}
139	}
140	if(\$hasilShuffle['data'][\$j]->playlist == 11) {
141	if(count(\$data11)<5&& \$dataIndex11 == \$f*5) {
142	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
143	array_push(\$data11, \$hasilShuffle['data'][\$j] );
144	}else{
145	\$dataIndex11++;
146	}
147	}
148	if(\$hasilShuffle['data'][\$j]->playlist == 12) {
149	if(count(\$data12)<5&& \$dataIndex12 == \$f*5) {

No	Source Code
150	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data12, \$hasilShuffle['data'][\$j]); } else { \$dataIndex12++; } } if (\$hasilShuffle['data'][\$j]->playlist == 13) { if(count(\$data13)<5&& \$dataIndex13 == \$f*5) { array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data13, \$hasilShuffle['data'][\$j]); } else { \$dataIndex13++; } } \$dataTesting = \$allDataTesting; \$jumlahBenar = array(0,0,0,0,0,0,0,0,0,0,0,0,0); \$jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0); for(\$j=0;\$j<count(\$dataTesting);\$j++) { \$hasilShuffle['dataTesting'][\$f][\$j] = \$dataTesting[\$j]; } \$hasilDistance = array(); for(\$k=0;\$k<count(\$hasilShuffle['data']);\$k++) { \$sama = 0; for(\$l=0; \$l<count(\$allDataTesting);\$l++) { if(\$allDataTesting[\$l]->no == \$hasilShuffle['data'][\$k]->no) { \$sama = 1; break; } } if(\$sama == 0) { \$kalkulasiDistance = new MyDistance(); \$kalkulasiDistance->no = \$hasilShuffle['data'][\$k]->no; \$totalPow = 0; for(\$p=0;\$p<\$pca;\$p++) { \$fitur = "pca".(\$p+1); \$totalPow += pow(abs(\$dataTesting[\$j]->\$fitur - \$hasilShuffle['data'][\$k]->\$fitur), 2); } } }

No	Source Code
186	}
187	\$kalkulasiDistance->distance = sqrt (\$totalPow);
188	array_push(\$hasilDistance, \$kalkulasiDistance);
189	}
190	}
191	usort (\$hasilDistance, "cmp");
192	for (\$k=0; \$k<\$nilaiK; \$k++) {
193	\$hasilShuffle['hasilDistance'][\$f][\$j][\$k] = \$hasilDistance[\$k];
194	}
195	\$hasilClass = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
196	for (\$k=0;\$k<\$nilaiK;\$k++) {
197	\$hasilClass[\$hasilShuffle['data'][\$hasilDistance[\$k]->no-1]->playlist]]+=1;
198	}
199	\$max = \$hasilClass[0];
200	\$classHigh = 0;
201	for (\$k=1;\$k<count (\$hasilClass);\$k++) {
202	if (\$max<\$hasilClass[\$k]) {
203	\$max = \$hasilClass[\$k];
204	\$classHigh = \$k;
205	}
206	}
207	\$hasilShuffle['dataHasilTesting'][\$f][\$j] = \$classHigh;
208	if (\$classHigh == \$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist) {
209	\$jumlahBenar[\$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist]]++;
210	} else {
211	\$jumlahSalah[\$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist]]++;
212	}
213	switch (\$hasilShuffle['data'][\$dataTesting[\$j]->no-1]->playlist) {
214	case 0:
215	\$dataHasil0[\$classHigh]]++;
216	break;
217	case 1:
218	\$dataHasil1[\$classHigh]]++;
219	break;
220	case 2:
221	\$dataHasil2[\$classHigh]]++;
222	break;

No	<i>Source Code</i>
	<pre> 223    case 3: 224        \$dataHasil3[\$classHigh]++; 225        break; 226    case 4: 227        \$dataHasil4[\$classHigh]++; 228        break; 229    case 5: 230        \$dataHasil5[\$classHigh]++; 231        break; 232    case 6: 233        \$dataHasil6[\$classHigh]++; 234        break; 235    case 7: 236        \$dataHasil7[\$classHigh]++; 237        break; 238    case 8: 239        \$dataHasil8[\$classHigh]++; 240        break; 241    case 9: 242        \$dataHasil9[\$classHigh]++; 243        break; 244    case 10: 245        \$dataHasil10[\$classHigh]++; 246        break; 247    case 11: 248        \$dataHasil11[\$classHigh]++; 249        break; 250    case 12: 251        \$dataHasil12[\$classHigh]++; 252        break; 253    case 13: 254        \$dataHasil13[\$classHigh]++; 255        break; 256    } 257    } 258    \$hasilShuffle["DataMatriks"][\$f][0] = \$dataHasil0; 259    \$hasilShuffle["DataMatriks"][\$f][1] = \$dataHasil1; 260    \$hasilShuffle["DataMatriks"][\$f][2] = \$dataHasil2; 261    \$hasilShuffle["DataMatriks"][\$f][3] = \$dataHasil3; </pre>

No	Source Code
262	\$hasilShuffle["DataMatriks"][\$f][4] = \$dataHasil4;
263	\$hasilShuffle["DataMatriks"][\$f][5] = \$dataHasil5;
264	\$hasilShuffle["DataMatriks"][\$f][6] = \$dataHasil6;
265	\$hasilShuffle["DataMatriks"][\$f][7] = \$dataHasil7;
266	\$hasilShuffle["DataMatriks"][\$f][8] = \$dataHasil8;
267	\$hasilShuffle["DataMatriks"][\$f][9] = \$dataHasil9;
268	\$hasilShuffle["DataMatriks"][\$f][10] = \$dataHasil10;
269	\$hasilShuffle["DataMatriks"][\$f][11] = \$dataHasil11;
270	\$hasilShuffle["DataMatriks"][\$f][12] = \$dataHasil12;
271	\$hasilShuffle["DataMatriks"][\$f][13] = \$dataHasil13;
272	\$calculateBenar = 0;
273	for(\$j=0; \$j<count(\$jumlahBenar); \$j++) {
274	\$calculateBenar += \$jumlahBenar[\$j];
275	}
276	\$hasilShuffle["HasilAkurasi"][\$f] = \$calculateBenar/70
277	*100;
278	}
279	\$this->load->view('ViewKlasifikasiPCAKNN', \$hasilShuffle);

### Lampiran C.3. Source Code PCA–MKNN Pengujian Kedua

No	Source Code
1	public function KlasifikasiPCAMKNN(\$pca,\$nilaiK) {
2	\$hasilShuffle['nilaiK'] = \$nilaiK;
3	\$hasilShuffle['indexPca'] = \$pca;
4	\$hasilShuffle['data'] = \$this->Data->ViewDataPca("data_pca".\$pca);
5	for(\$g=0; \$g<10; \$g++) {
6	\$data0 = array();
7	\$data1 = array();
8	\$data2 = array();
9	\$data3 = array();
10	\$data4 = array();
11	\$data5 = array();
12	\$data6 = array();
13	\$data7 = array();
14	\$data8 = array();
15	\$data9 = array();
16	\$data10 = array();
17	\$data11 = array();

No	Source Code
18	\$data12 = array();
19	\$data13 = array();
20	\$allDataTesting = array();
21	//
22	\$dataIndex0 = 0;
23	\$dataIndex1 = 0;
24	\$dataIndex2 = 0;
25	\$dataIndex3 = 0;
26	\$dataIndex4 = 0;
27	\$dataIndex5 = 0;
28	\$dataIndex6 = 0;
29	\$dataIndex7 = 0;
30	\$dataIndex8 = 0;
31	\$dataIndex9 = 0;
32	\$dataIndex10 = 0;
33	\$dataIndex11 = 0;
34	\$dataIndex12 = 0;
35	\$dataIndex13 = 0;
36	//
37	\$hasilData0 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
38	\$hasilData1 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
39	\$hasilData2 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
40	\$hasilData3 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
41	\$hasilData4 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
42	\$hasilData5 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
43	\$hasilData6 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
44	\$hasilData7 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
45	\$hasilData8 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
46	\$hasilData9 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
47	\$hasilData10 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
48	\$hasilData11 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
49	\$hasilData12 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
50	\$hasilData13 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
51	for(\$j=0;\$j<count(\$hasilShuffle['data']);\$j++) {
52	if(\$hasilShuffle['data'][\$j]->playlist == 0){
53	if(count(\$data0)<5&& \$dataIndex0 == \$g*5){
54	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
55	array_push(\$data0, \$hasilShuffle['data'][\$j] );
56	}else{

No	Source Code
57	\$dataIndex0++;
58	}
59	}
60	if(\$hasilShuffle['data'][\$j]->playlist == 1){
61	if(count(\$data1)<5&& \$dataIndex1 == \$g*5){
62	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
63	array_push(\$data1, \$hasilShuffle['data'][\$j]) );
64	}else{
65	\$dataIndex1++;
66	}
67	}
68	if(\$hasilShuffle['data'][\$j]->playlist == 2){
69	if(count(\$data2)<5&& \$dataIndex2 == \$g*5){
70	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
71	array_push(\$data2, \$hasilShuffle['data'][\$j]) );
72	}else{
73	\$dataIndex2++;
74	}
75	}
76	if(\$hasilShuffle['data'][\$j]->playlist == 3){
77	if(count(\$data3)<5&& \$dataIndex3 == \$g*5){
78	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
79	array_push(\$data3, \$hasilShuffle['data'][\$j]) );
80	}else{
81	\$dataIndex3++;
82	}
83	}
84	if(\$hasilShuffle['data'][\$j]->playlist == 4){
85	if(count(\$data4)<5&& \$dataIndex4 == \$g*5){
86	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
87	array_push(\$data4, \$hasilShuffle['data'][\$j]) );
88	}else{
89	\$dataIndex4++;
90	}
91	}
92	if(\$hasilShuffle['data'][\$j]->playlist == 5){
93	if(count(\$data5)<5&& \$dataIndex5 == \$g*5){

No	Source Code
94	array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data5, \$hasilShuffle['data'][\$j]); } else { \$dataIndex5++; } } if(\$hasilShuffle['data'][\$j]->playlist == 6) { if(count(\$data6)<5&& \$dataIndex6 == \$g*5) { array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data6, \$hasilShuffle['data'][\$j]); } else { \$dataIndex6++; } } if(\$hasilShuffle['data'][\$j]->playlist == 7) { if(count(\$data7)<5&& \$dataIndex7 == \$g*5) { array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data7, \$hasilShuffle['data'][\$j]); } else { \$dataIndex7++; } } if(\$hasilShuffle['data'][\$j]->playlist == 8) { if(count(\$data8)<5&& \$dataIndex8 == \$g*5) { array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data8, \$hasilShuffle['data'][\$j]); } else { \$dataIndex8++; } } if(\$hasilShuffle['data'][\$j]->playlist == 9) { if(count(\$data9)<5&& \$dataIndex9 == \$g*5) { array_push(\$allDataTesting, \$hasilShuffle['data'][\$j]); array_push(\$data9, \$hasilShuffle['data'][\$j]); } else { \$dataIndex9++; } }

No	Source Code
130	}
131	}
132	if(\$hasilShuffle['data'][\$j]->playlist == 10) {
133	if(count(\$data10)<5&& \$dataIndex10 == \$g*5){
134	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
135	array_push(\$data10, \$hasilShuffle['data'][\$j ]);
136	}else{
137	\$dataIndex10++;
138	}
139	}
140	if(\$hasilShuffle['data'][\$j]->playlist == 11) {
141	if(count(\$data11)<5&& \$dataIndex11 == \$g*5){
142	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
143	array_push(\$data11, \$hasilShuffle['data'][\$j ]);
144	}else{
145	\$dataIndex11++;
146	}
147	}
148	if(\$hasilShuffle['data'][\$j]->playlist == 12) {
149	if(count(\$data12)<5&& \$dataIndex12 == \$g*5){
150	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
151	array_push(\$data12, \$hasilShuffle['data'][\$j ]);
152	}else{
153	\$dataIndex12++;
154	}
155	}
156	if(\$hasilShuffle['data'][\$j]->playlist == 13) {
157	if(count(\$data13)<5&& \$dataIndex13 == \$g*5){
158	array_push(\$allDataTesting, \$hasilShuffle['d ata'][\$j]);
159	array_push(\$data13, \$hasilShuffle['data'][\$j ]);
160	}else{
161	\$dataIndex13++;
162	}
163	}
164	}
165	\$dataTesting = \$allDataTesting;

No	Source Code
166	\$jumlahBenar = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
167	\$jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
168	for(\$j=0;\$j<count(\$dataTesting);\$j++) {
169	\$hasilShuffle['dataTesting'][\$g][\$j] = \$dataTesting[\$j];
170	\$hasilDistance = array();
171	for(\$k=0;\$k<count(\$hasilShuffle['data']);\$k++) {
172	\$sama = 0;
173	for(\$l=0; \$l<count(\$allDataTesting);\$l++) {
174	if(\$allDataTesting[\$l] - >no == \$hasilShuffle['data'][\$k] - >no) {
175	\$sama = 1;
176	break;
177	}
178	}
179	if(\$sama == 0) {
180	\$kalkulasiDistance = new MyDistance();
181	\$kalkulasiDistance->no = \$hasilShuffle['data'][\$k]->no;
182	\$totalPow = 0;
183	for(\$p=0;\$p<\$pca;\$p++) {
184	\$fitur = "pca".(\$p+1);
185	\$totalPow += pow(abs(\$dataTesting[\$j] - >\$fitur - \$hasilShuffle['data'][\$k]->\$fitur), 2);
186	}
187	\$kalkulasiDistance->distance = sqrt(\$totalPow);
188	array_push(\$hasilDistance, \$kalkulasiDistance);
189	}
190	}
191	usort(\$hasilDistance, "cmp");
192	for(\$k=0; \$k<\$nilaiK; \$k++) {
193	\$hasilShuffle['hasilDistance'][\$g][\$j][\$k] = \$hasilDistance[\$k];
194	}
195	\$weightResult = array();
196	for(\$i=0;\$i<\$nilaiK;\$i++) {
197	\$dataTest = \$hasilShuffle['data'][\$hasilDistance[\$i]->no-1];
198	\$hasilShuffle['dataTestingResult'][\$g][\$j][\$i] = \$dataTest;
199	\$resultDistance = array();
200	for(\$k=0;\$k<count(\$hasilShuffle['data']);\$k++) {
201	\$sama = 0;
202	for(\$l=0; \$l<count(\$allDataTesting);\$l++) {

No	Source Code
203	if(\$allDataTesting[\$l]->no == \$hasilShuffle['data'][\$k]->no){
204	\$sama = 1;
205	break;
206	}
207	}
208	if(\$sama == 0 && \$dataTest->no != \$hasilShuffle['data'][\$k]->no){
209	\$kalDistance = new MyDistance();
210	\$kalDistance->no = \$hasilShuffle['data'][\$k]->no;
211	\$totalPow = 0;
212	for(\$p=0;\$p<\$pca;\$p++){
213	\$fitur = "pca".(\$p+1);
214	\$totalPow += pow(abs(\$dataTest->\$fitur) - \$hasilShuffle['data'][\$k]->\$fitur), 2);
215	}
216	\$kalkulasiDistance->distance = sqrt(\$totalPow);
217	array_push(\$resultDistance, \$kalDistance);
218	}
219	}
220	usort(\$resultDistance, "cmp");
221	for(\$k=0; \$k<\$nilaiK; \$k++) {
222	\$hasilShuffle['resultDistance'][\$g][\$j][\$i][\$k] = \$resultDistance[\$k];
223	}
224	\$validasi = 0;
225	for(\$k=0; \$k<\$nilaiK; \$k++) {
226	if(\$hasilShuffle['data'][\$resultDistance[\$k]->no-1]->playlist == \$dataTest->playlist) {
227	\$validasi++;
228	}
229	}
230	\$validasi = \$validasi/\$nilaiK;
231	\$hasilShuffle['Validasi'][\$g][\$j][\$i] = \$validasi;
232	\$cacheWeight = \$validasi * (1/(\$hasilDistance[\$i]->distance+0.5));
233	\$hasilShuffle['Weight'][\$g][\$j][\$i] = \$cacheWeight;
234	\$classWight = new MyWeight();
235	\$classWight->playlist = \$dataTest->playlist;
236	\$classWight->weight = \$cacheWeight;
237	array_push(\$weightResult, \$classWight);
238	}

No	Source Code
239	\$hasilClassDistance = array(0,0,0,0,0,0,0,0,0,0,0,0,0, 240 ,0,0,0); 241 for (\$k=0;\$k<count(\$weightResult);\$k++){ 242 \$hasilClassDistance[\$weightResult[\$k]- 243 >playlist]+=\$weightResult[\$k]->weight; 244 } 245 \$chooseClass = 0; 246 \$cacheMax = 0; 247 for (\$k=0;\$k<count(\$hasilClassDistance);\$k++){ 248 if (\$cacheMax<\$hasilClassDistance[\$k]) { 249 \$cacheMax = \$hasilClassDistance[\$k]; 250 \$chooseClass = \$k; 251 } 252 \$hasilShuffle['dataHasilTesting'][\$g][\$j] = \$chos eClass; 253 if(\$chooseClass == \$hasilShuffle['data'][\$dataTest 254 ing[\$j]->no-1]->playlist){ 255 \$jumlahBenar[\$hasilShuffle['data'][\$dataTestin 256 g[\$j]->no-1]->playlist]++; 257 }else { 258 \$jumlahSalah[\$hasilShuffle['data'][\$dataTestin 259 g[\$j]->no-1]->playlist]++; 260 } 261 switch (\$hasilShuffle['data'][\$dataTesting[\$j]->no- 262 1]->playlist) { 263 case 0: 264 \$hasilData0[\$chooseClass]++; 265 break; 266 case 1: 267 \$hasilData1[\$chooseClass]++; 268 break; 269 case 2: 270 \$hasilData2[\$chooseClass]++; 271 break; 272 case 3: 273 \$hasilData3[\$chooseClass]++; 274 break; 275 case 4: 276 \$hasilData4[\$chooseClass]++; 277 break; 278 case 5: 279 \$hasilData5[\$chooseClass]++; 280 break; 281 case 6:

No	<i>Source Code</i>
277	\$hasilData6[\$chooseClass]++; break;
278	case 7: \$hasilData7[\$chooseClass]++; break;
279	case 8: \$hasilData8[\$chooseClass]++; break;
280	case 9: \$hasilData9[\$chooseClass]++; break;
281	case 10: \$hasilData10[\$chooseClass]++; break;
282	case 11: \$hasilData11[\$chooseClass]++; break;
283	case 12: \$hasilData12[\$chooseClass]++; break;
284	case 13: \$hasilData13[\$chooseClass]++; break;
285	}
286	}
287	\$hasilShuffle["DataMatriks"][\$g][0] = \$hasilData0; \$hasilShuffle["DataMatriks"][\$g][1] = \$hasilData1; \$hasilShuffle["DataMatriks"][\$g][2] = \$hasilData2; \$hasilShuffle["DataMatriks"][\$g][3] = \$hasilData3; \$hasilShuffle["DataMatriks"][\$g][4] = \$hasilData4; \$hasilShuffle["DataMatriks"][\$g][5] = \$hasilData5; \$hasilShuffle["DataMatriks"][\$g][6] = \$hasilData6; \$hasilShuffle["DataMatriks"][\$g][7] = \$hasilData7; \$hasilShuffle["DataMatriks"][\$g][8] = \$hasilData8; \$hasilShuffle["DataMatriks"][\$g][9] = \$hasilData9; \$hasilShuffle["DataMatriks"][\$g][10] = \$hasilData10; \$hasilShuffle["DataMatriks"][\$g][11] = \$hasilData11; \$hasilShuffle["DataMatriks"][\$g][12] = \$hasilData12; \$hasilShuffle["DataMatriks"][\$g][13] = \$hasilData13;
288	\$calculateBenar = 0;
289	
290	
291	
292	
293	
294	
295	
296	
297	
298	
299	
300	
301	
302	
303	
304	
305	
306	
307	
308	
309	
310	
311	
312	
313	
314	
315	
316	

No	Source Code
317	for (\$j=0; \$j<count(\$jumlahBenar); \$j++) {
318	\$calculateBenar += \$jumlahBenar[\$j];
319	}
320	\$hasilShuffle["HasilAkurasi"][\$g] = \$calculateBenar/70 *100;
321	}
322	\$this->load->view('ViewKlasifikasiPCAMKNN', \$hasilShuffle);
323	}

## Lampiran C.4. Implementasi Sistem pada PCA-KNN dan PCA-MKNN Pengujian Kedua

### 1. Output Input Reduksi Jumlah Fitur PCA

Pada Gambar 1, Pengguna memasukkan jumlah fitur direduksi dengan PCA, tapi nilai inputan tidak boleh 0.

PROGRAM STUDI INFORMATIKA
Normalisasi
Klasifikasi
Tambah PCA
Klasifikasi dengan PCA

**KLASIFIKASI MUSIK BERDASARKAN GENRE PADA LAYANAN STREAMING MUSIK SPOTIFY MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR DAN MODIFIED K-NEAREST NEIGHBOR**

**Reduksi Fitur Principal Component Analysis (PCA)**

		11							
		<input type="button" value="Submit"/>							
<small>* Input jumlah fitur tidak boleh 0</small>									
Fitur		Eigenvalue	ComputePercentageExplained						
Danceability		17.324267064992977	0.31452914321125314						
Energy		13.89250122875269	0.5667531762481529						
Key		6.244412386600236	0.6801230331342691						
Loudness		5.732648412121658	0.7842016060765816						
Mode		3.485010144691698	0.8474733841405						
Speechiness		2.689476677515101	0.8963019357858931						
Acoustickness		1.9947499148894556	0.9325174350794733						
Instrumentalness		1.297703111163251	0.95607776500082						
Liveness		0.9710438097208153	0.9737074619078391						
Valance		0.86763196747731051	0.9859863217894871						
Tempo		0.514675773253626	0.9953304706340654						
Duration		0.25719705436473	1						
No	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5	Fitur 6	Fitur 7	Fitur 8	Fitur 9
1	0.2312645772748398	-0.4827115441075232	-0.520194980024521	0.012135439629351595	0.47396877131834597	-0.4123327086292669	0.07734414734617091	-0.1495672863757832	-0.0638116898179641
2	0.01934240277437869	-0.3747529627271094	0.5757053015166146	-0.24897089425376207	0.08015894457437776	-0.001109306742370537	0.034268016441435976	0.06019320322426369	0.0586020220858475
3	0.083597273810852	0.6841549359522756	0.20225366258972038	0.3762425626464432	-0.037488648841577085	0.1832789504489079	-0.4605257211878374	-0.0830638131978715	0.1129897505799829
4	0.3271277734171446	-0.4811931365663529	-0.18496728090627523	-0.3085874191211468	-0.03802834431011999	0.1544283950574414	0.23705123693466046	0.006743015115632442	0.0447734762535670
5	0.15598588644652964	0.6678102606565228	0.2396228293056286	0.22409086665619757	0.06638330765520668	0.043486059991553555	-0.05910498001604391	0.22473049527106423	0.035130503801711
6	0.21319816694594165	-0.37235190979872157	0.35274655787795217	0.10677856206211098	-0.03699126848265883	0.1993068131921558	0.030678862189735936	-0.2674374996770475	0.2731672811628923
7	-0.22342857886450446	-0.292912147774255	0.545976798189593	0.12056759889527662	0.2730975628975993	0.08316097537653386	0.0396756875876561	0.10505776508487226	0.0118696658541280
8	0.3000485298330027	-0.4705804998006311	-0.39230900334848884	-0.03512793638979081	-0.11166704773536885	0.1940910203002771	-0.09591855250640476	-0.003037061595518289	-0.06475988913294
9	-0.6708777567462352	-0.1996807725859794	0.3934970742711429	0.473913404677113	0.33814644734854016	0.370183115311652	-0.32330174626028113	0.010634265082262648	0.0846068247896860
10	0.41043826641737385	-0.5144572957352325	-0.4220973194390545	-0.2226747913113149	0.059702858599730214	-0.1593540963842322	0.09435501243938998	0.14520369735505267	-0.055190628084343
11	-0.2535219227581848	-0.4106703544367904	-0.26650399882319914	-0.3841951915351323	-0.44226787503557946	-0.16108723904928662	0.43782615436313055	-0.15041868141457887	0.1841828373169988
12	-0.10885031232148373	-0.35925443383045	-0.1665306724368784	0.3365898293038452	0.6645098673463474	-0.3544931586357392	0.12275432119121948	-0.11420927182509048	0.0715832874039704
13	0.4414658505726135	0.5482592494905898	-0.29639826773520095	-0.06948528084151075	0.1940457174556187	-0.10617392593344725	0.3650235488280957	0.23307105342420367	0.0118990428187101
14	0.43025285922891865	-0.5185267237616005	-0.5920776759706625	0.056192615640904754	0.26065902605530805	-0.522887298334849	-0.1471069781799442	-0.07139981380771727	-0.020750051054564

Gambar 1. Input Jumlah Fitur direduksi PCA

## 2. Output Input Klasifikasi

Pengguna memasukkan nilai  $k$  yang nantinya akan digunakan untuk proses PCA–KNN dan PCA–MKNN. Nilai  $k$  dibatasi hanya bilangan asli (bilangan bulat positif yang bukan 0), nilai  $k$  tidak boleh genap, melainkan harus bilangan ganjil untuk menghindari kemungkinan dua kelas berbeda memiliki jumlah suara yang sama dan memasukkan jumlah fitur direduksi dengan PCA tidak boleh 0. Form dari input nilai  $k$  seperti pada Gambar 2.

Gambar 2. Input Nilai  $k$

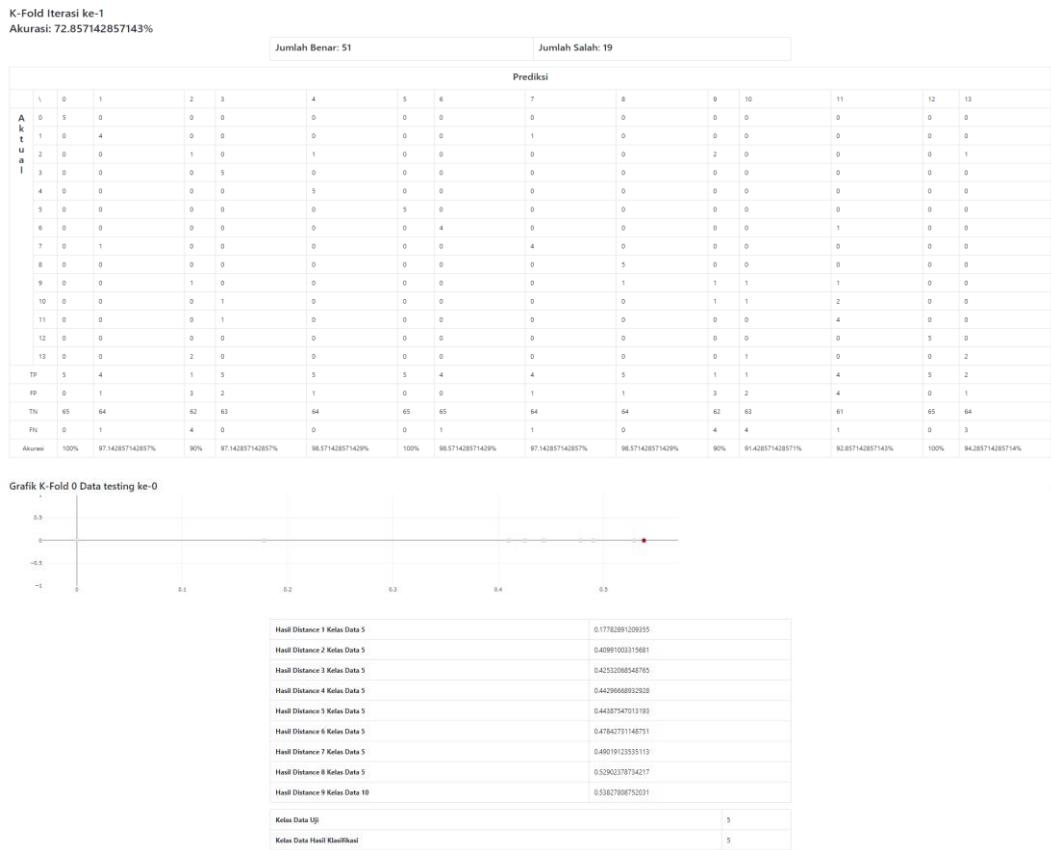
## 3. Output Klasifikasi PCA–KNN

Output dari PCA–KNN akan munjur jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 2, merupakan hasil dari klasifikasi PCA–KNN.

Np	Fitur PCA1	Fitur PCA2	Fitur PCA3	Fitur PCA4	Fitur PCA5	Fitur PCA6	Fitur PCA7	Fitur PCA8	Fitur PCA9	Fitur PCA10	Fitur PCA11
1	0.2312645772746398	-0.48271544107532	0.0121354962951195	0.4739687731834937	-0.4323327086292669	0.07734414734817091	-0.495872883757832	-0.0683168981794408	0.04715941334418224	0.048079931434268	-0.2056489013436584
2	0.093424027747898	-0.3747529627271094	0.575703015166146	-0.24697092437367207	0.0801589457437176	-0.001109306742370537	0.03420810044143957	0.00019130312434369	0.05686020208547176	-0.20091399723401	-0.2056489013436584
3	0.08395727810852	0.648145935275276	0.2023584269723072	0.376425426464442	-0.03748084884157705	0.1832789504880279	0.0465537216787374	-0.030308811978715	0.11289755794926	-0.02064601240337083	-0.2377772090642742
4	0.327177774471446	-0.4811933155643529	-0.1849673809672523	-0.3093874191214417	-0.05902834431011999	0.1544283955754714	0.273705123893464049	0.007430111563342	0.04477474525357678	0.232447053511552	-0.0572714203027406179
5	0.15998580044652964	0.6870102060545228	0.22962829232056286	0.2240806664619757	0.0683830705520668	0.0424865599153555	-0.05910498001604391	0.2247304827106423	0.035130503007171185	-0.01326048901410916	0.018492699532465757
6	0.2317981664954165	-0.37235190979872157	0.35274655787179517	0.106785262611998	-0.039912684265883	0.195906131921558	0.05078661897339598	-0.074749674674747	0.2731872816299275	0.023180687871031727	-0.1581476061060067
7	-0.2324285788450446	-0.39293214777425	0.545075679818993	0.1205679068372762	0.2730975630975993	0.0831607731785386	0.0398756683785361	0.120517853847226	0.0118966645413802	-0.12020149250303007	0.067231147448584
8	0.0000452893310027	-0.4705649890000311	-0.3923090334484884	-0.0312796393979081	-0.11166740771536885	0.19409102002002771	-0.0999185520640476	-0.0030370159518389	-0.064799995112427	0.066244274534672	0.0650021708447153
9	-0.670977754762352	-0.19960772585974	0.3934970742711429	0.4797034046771113	0.3384644734545016	0.370180115111652	-0.323307146265028113	0.01063426508262648	0.0846065279868608	0.01205971562449323	0.0574682296954374
10	0.1043826641773785	-0.514437297532329	-0.420297194980545	-0.2226747913113149	0.097028589730214	-0.1190548093842322	0.0943501243938986	0.1432039973505267	-0.0511906288434332	0.03813800926000771	0.082777070849447

Gambar 2. Output Klasifikasi PCA–KNN

Output dari PCA–KNN akan munjur jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 3, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-1.



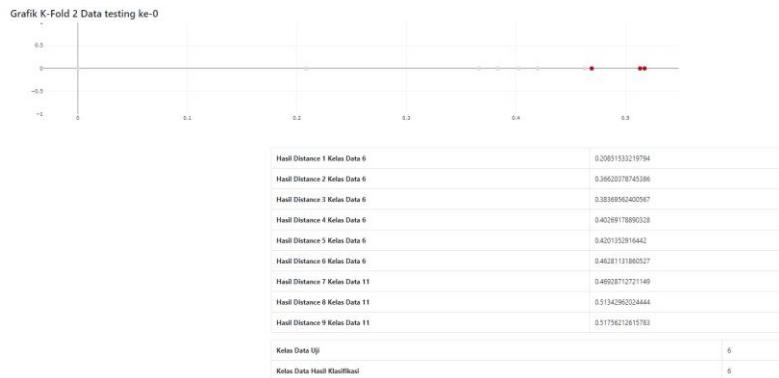
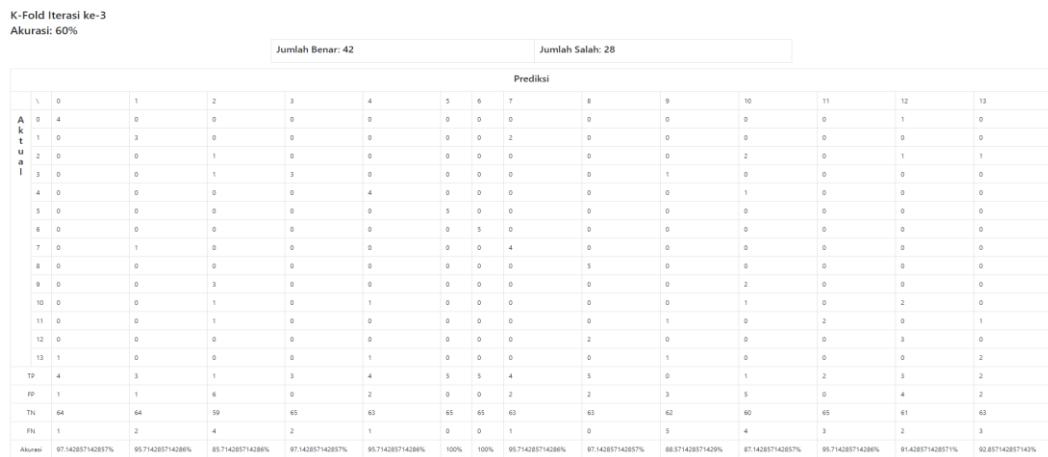
Gambar 3. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-1  
Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 4, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-2.





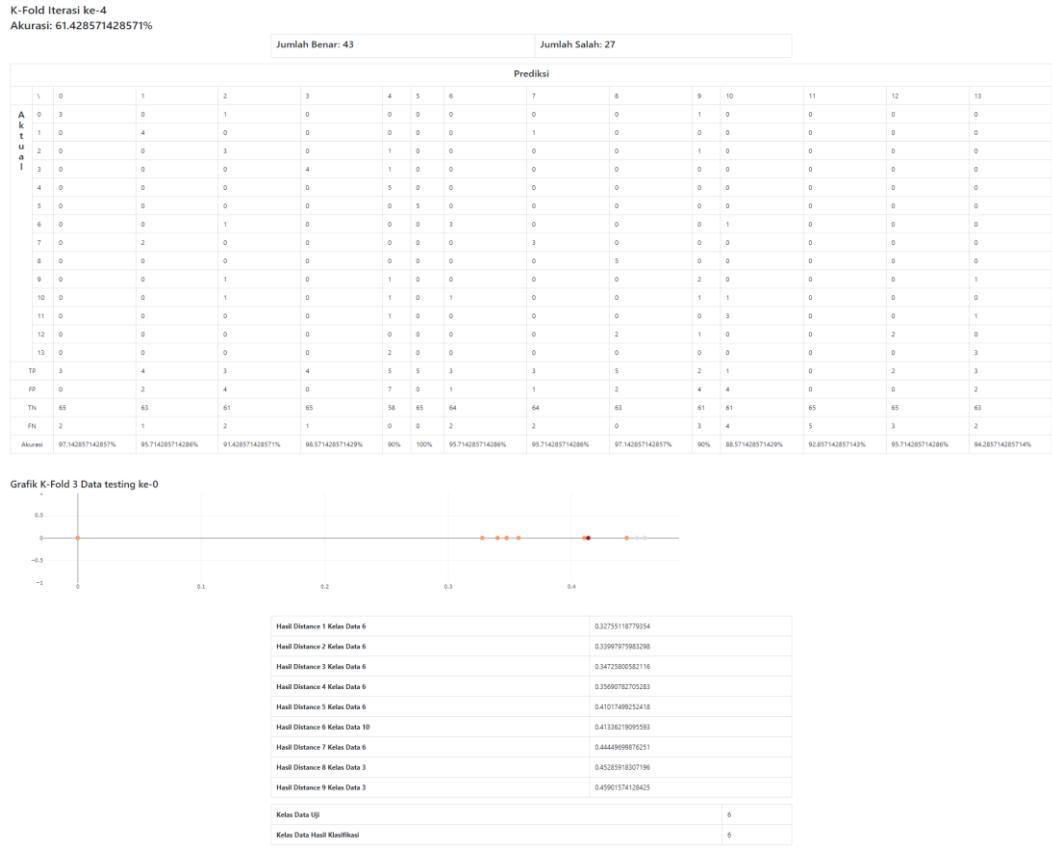
Gambar 4. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-2

Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 5, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-3.



Gambar 5. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-3

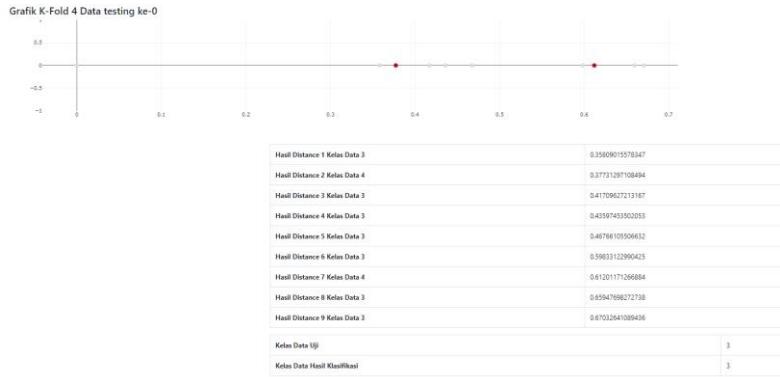
Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 6, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-4.



Gambar 6. Output Klasifikasi PCA-KNN K-Fold Iterasi ke-4

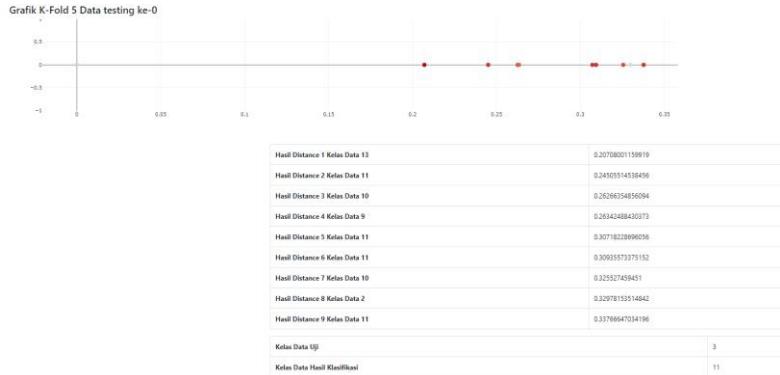
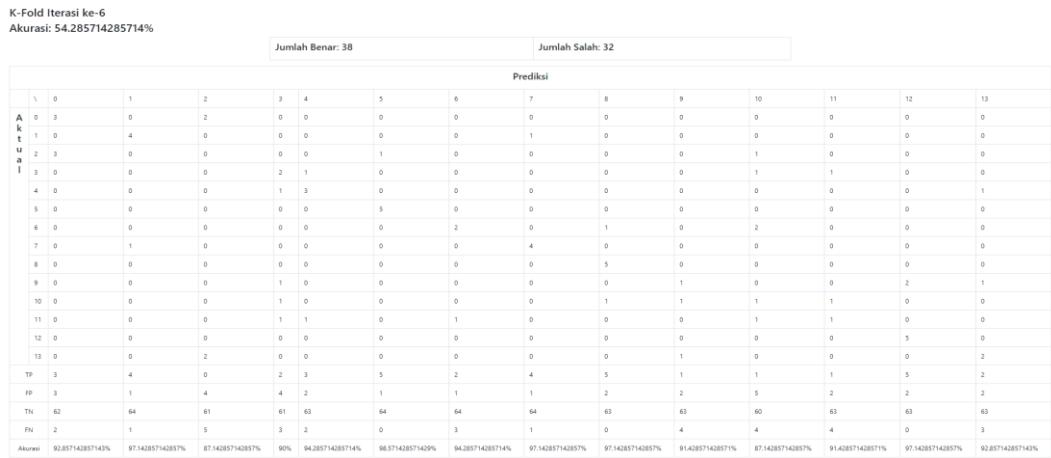
Output dari PCA-KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 7, merupakan hasil dari klasifikasi PCA-KNN K-Fold Iterasi ke-5.





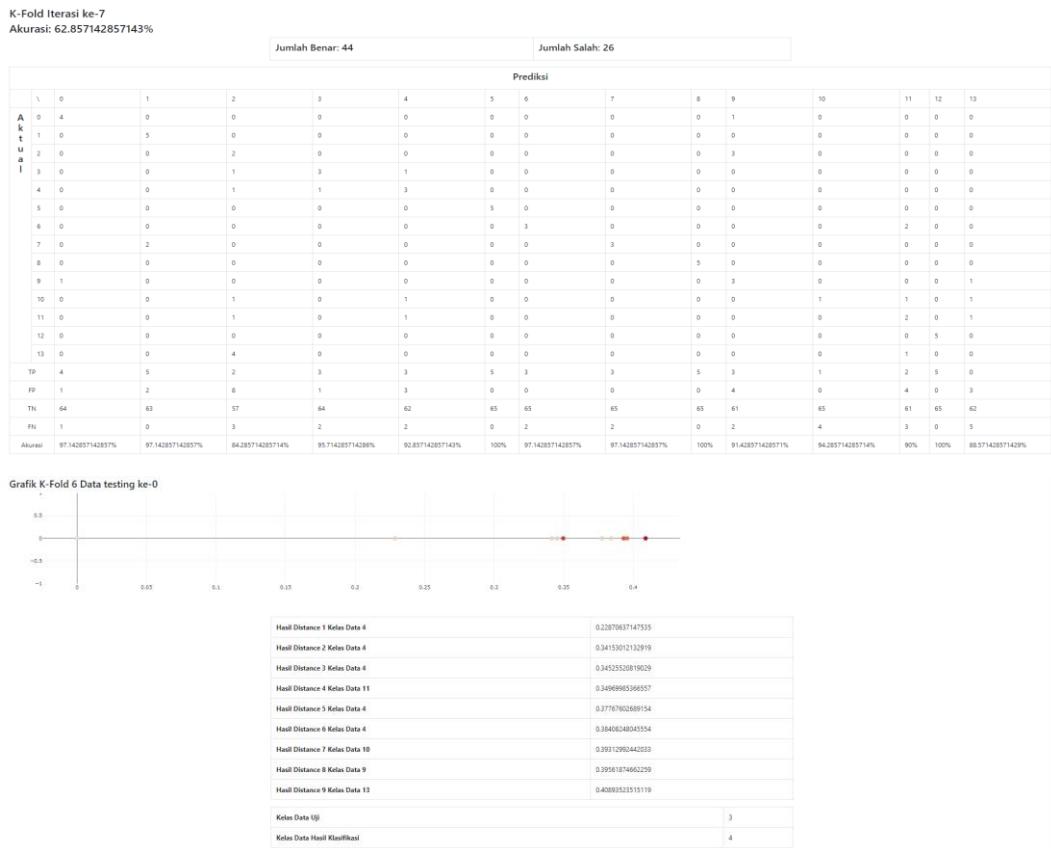
Gambar 7. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-5

Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 8, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-6.



Gambar 8. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-6

Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 9, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-7.



Gambar 9. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-7

Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 10, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-8.

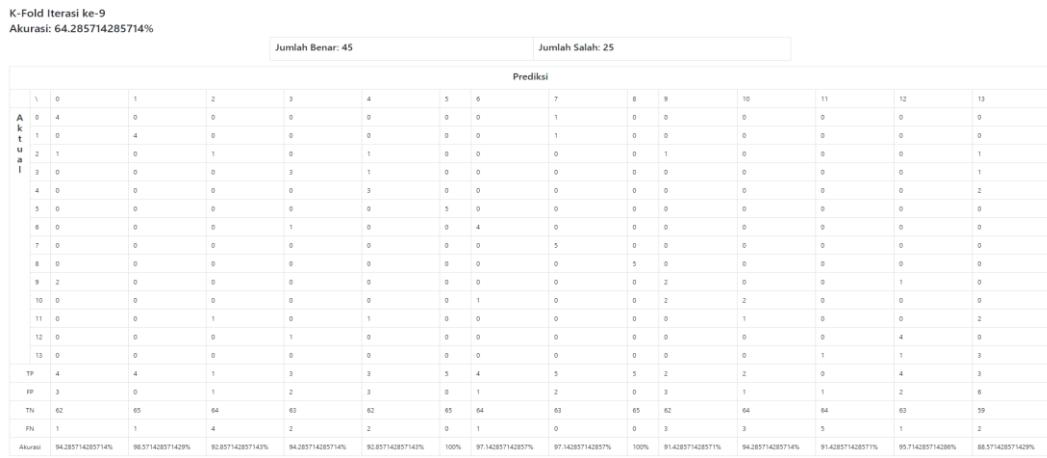
K-Fold Iterasi ke-8  
Akurasi: 47.142857142857%

Prediksi														
\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	0	1	0	0	0
k	1	0	1	0	0	0	0	4	0	0	0	0	0	0
t	2	1	0	2	0	0	0	0	0	1	0	0	1	0
u	3	0	0	0	3	0	0	0	0	1	0	0	1	0
a	4	0	0	0	0	4	0	0	0	0	1	0	0	0
i	5	0	0	0	0	0	4	0	0	0	0	0	1	0
l	6	0	0	0	0	0	0	2	0	0	1	1	1	0
	7	0	3	0	0	0	0	0	2	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	4	0	0	1	0
	9	1	0	1	0	0	0	0	0	2	0	0	0	0
	10	0	0	0	2	1	0	0	0	0	0	1	0	1
	11	0	0	1	0	1	0	0	0	0	1	2	0	0
	12	0	0	0	1	0	0	0	0	0	1	0	3	0
	13	0	0	0	1	3	0	0	0	0	1	0	0	0
TP	4	1	2	3	4	4	2	2	4	2	0	2	3	0
FP	2	3	2	5	5	0	0	4	0	3	4	3	5	1
TN	63	62	63	60	60	65	65	61	65	62	61	62	60	64
FN	1	4	3	2	1	1	3	2	1	3	3	2	5	1
Akurasi	95.714285714288%	90%	92.857142857143%	90%	91.428571428571%	95.571428571429%	95.714285714288%	90%	98.571428571429%	91.428571428571%	97.142857142857%	90%	91.428571428571%	



Gambar 10. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-8

Output dari PCA–KNN akan munjur jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 11, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-9.



Gambar 11. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-9

Output dari PCA–KNN akan munjur jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 12, merupakan hasil dari klasifikasi PCA–KNN K-Fold Iterasi ke-10.



Gambar 12. Output Klasifikasi PCA–KNN K-Fold Iterasi ke-10

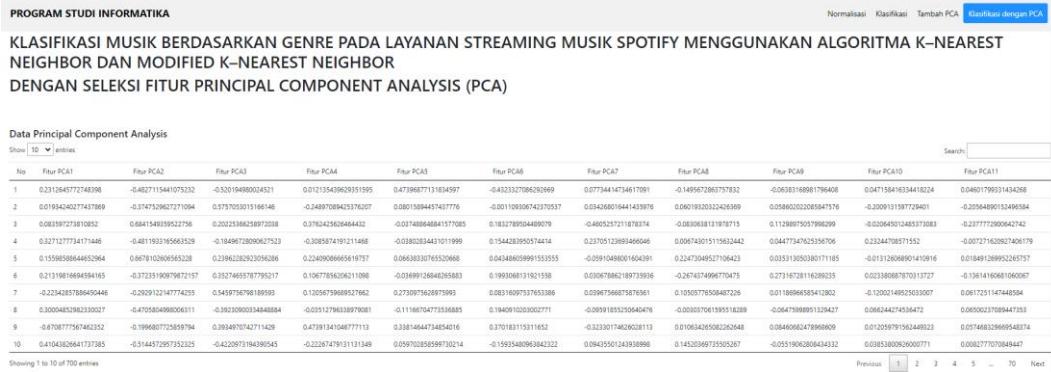
Output dari PCA–KNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 13, merupakan hasil dari rata–rata akurasi semua K-Fold klasifikasi PCA–KNN.

Total Rata-Rata Akurasi Semua k-fold: 60.285714285714%

Gambar 13. Output Rata–Rata Akurasi Semua K-Fold Klasifikasi PCA–KNN

#### 4. Output Klasifikasi PCA–MKNN

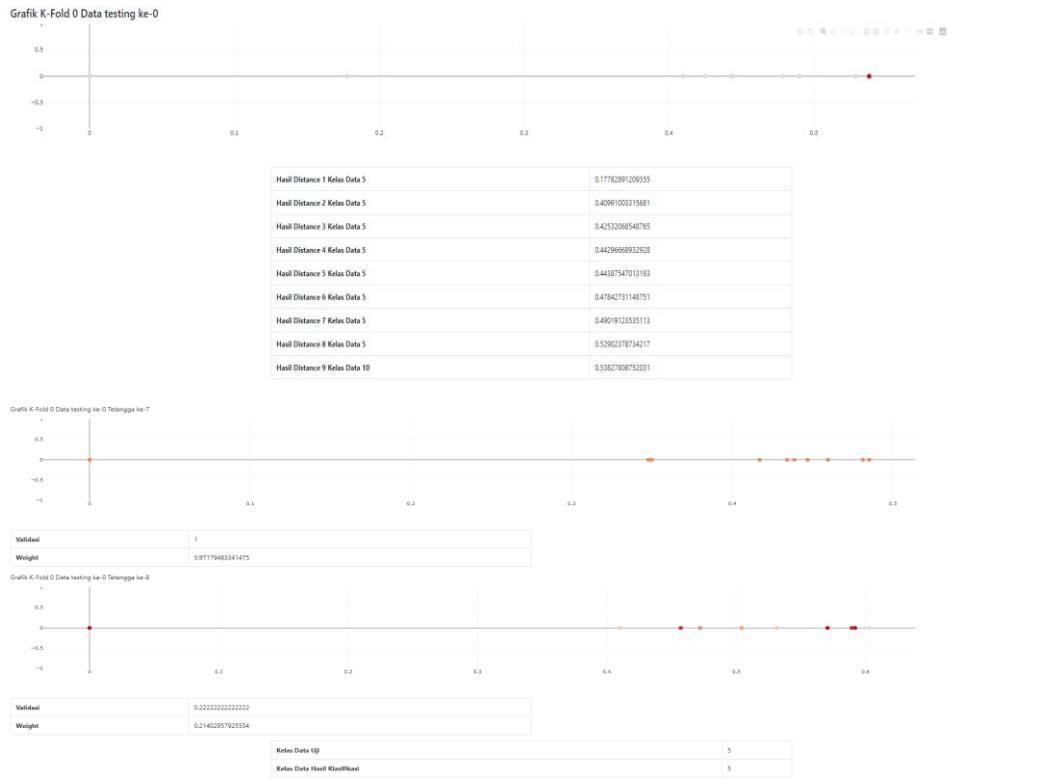
Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 14, merupakan hasil dari klasifikasi PCA–MKNN.



Gambar 14. Output Klasifikasi PCA–MKNN

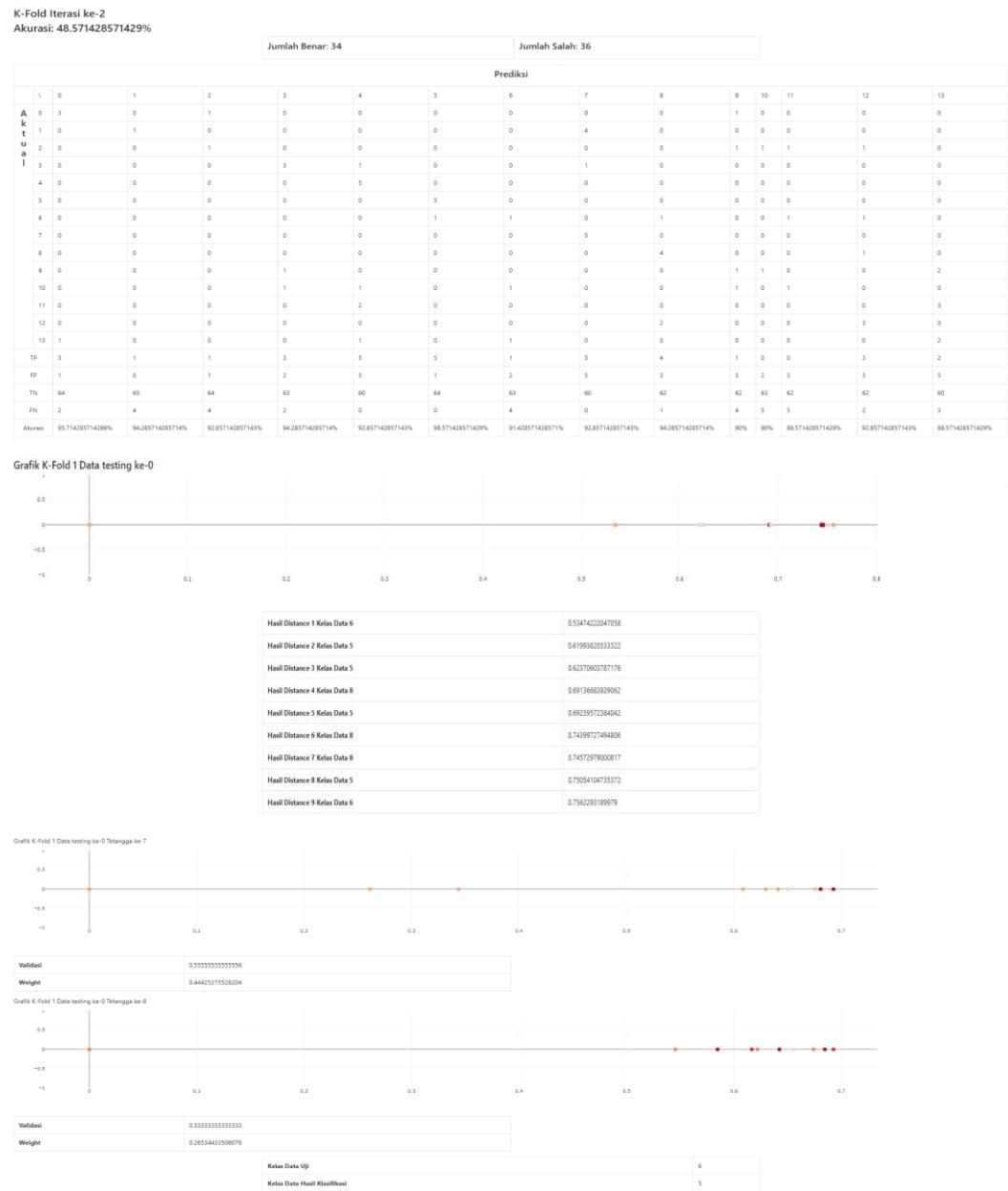
Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 15, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-1.

Prediksi														
\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	5	0	0	0	0	0	0	0	0	0	0	0	0
k	1	0	3	0	0	0	0	0	2	0	0	0	0	0
t	2	1	0	0	0	1	0	0	0	2	0	0	0	1
u	3	0	0	0	3	0	0	0	0	0	0	1	1	0
a	4	0	0	0	0	5	0	0	0	0	0	0	0	0
l	5	0	0	0	0	0	5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	5	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	4	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	5	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	2	1	1	0
10	0	0	0	1	0	0	0	0	0	0	1	0	3	0
11	0	0	0	0	1	0	0	0	0	0	0	0	3	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	2	0	0	0	0	0	0	0	2
TB	5	3	0	3	5	5	5	4	5	2	0	3	5	2
FP	1	1	1	1	4	0	0	2	1	3	1	5	2	1
TN	64	64	64	64	61	65	65	63	64	62	64	60	63	64
FN	0	2	5	2	0	0	0	1	0	3	5	2	0	3
Akurasi	95.7714285714285%	95.714285714285%	91.428571428571%	95.714285714285%	94.285714285714%	100%	100%	95.714285714285%	95.714285714285%	91.428571428571%	91.428571428571%	90%	97.142857142857%	94.285714285714%



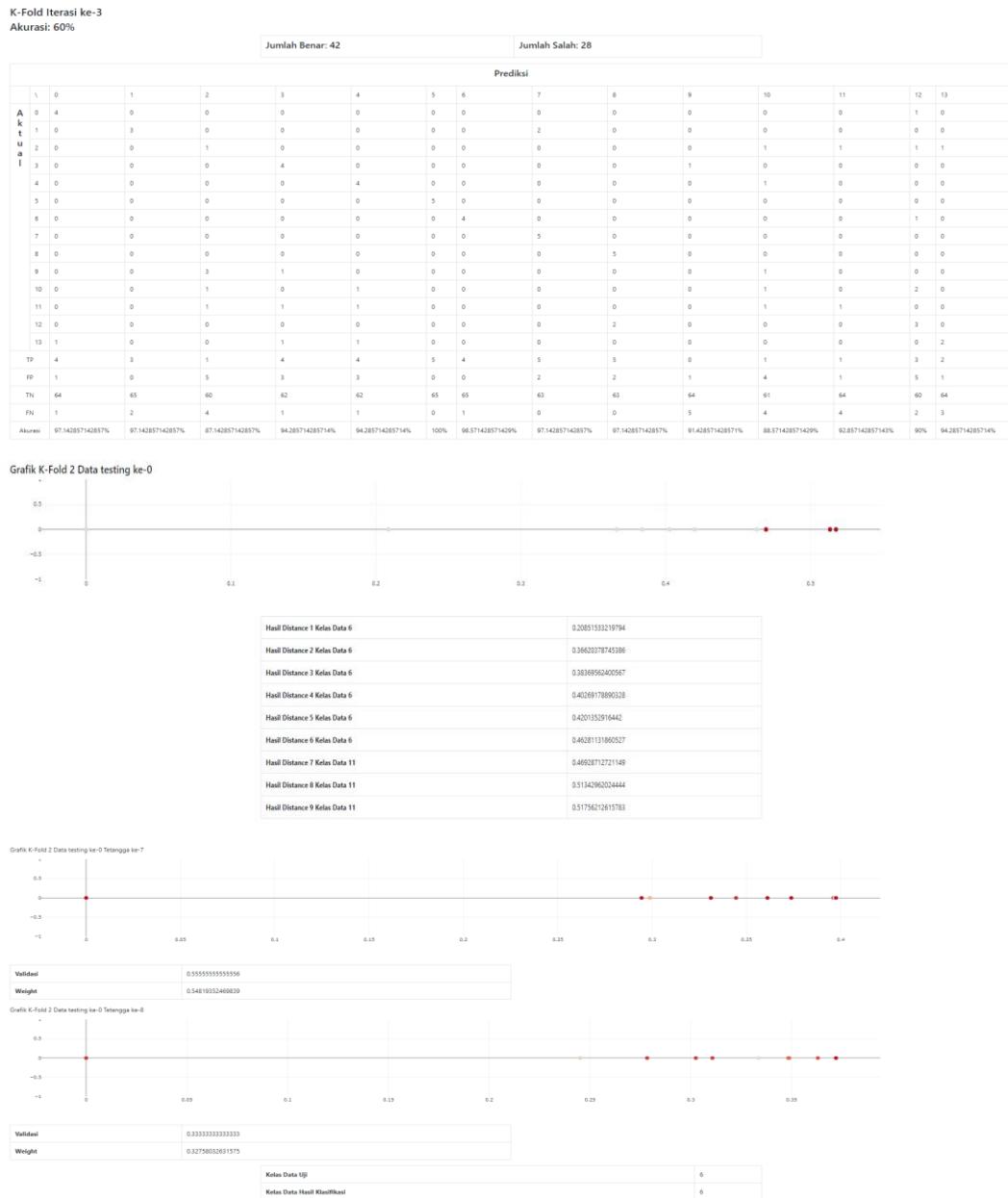
Gambar 15. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-1

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 16, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-2.



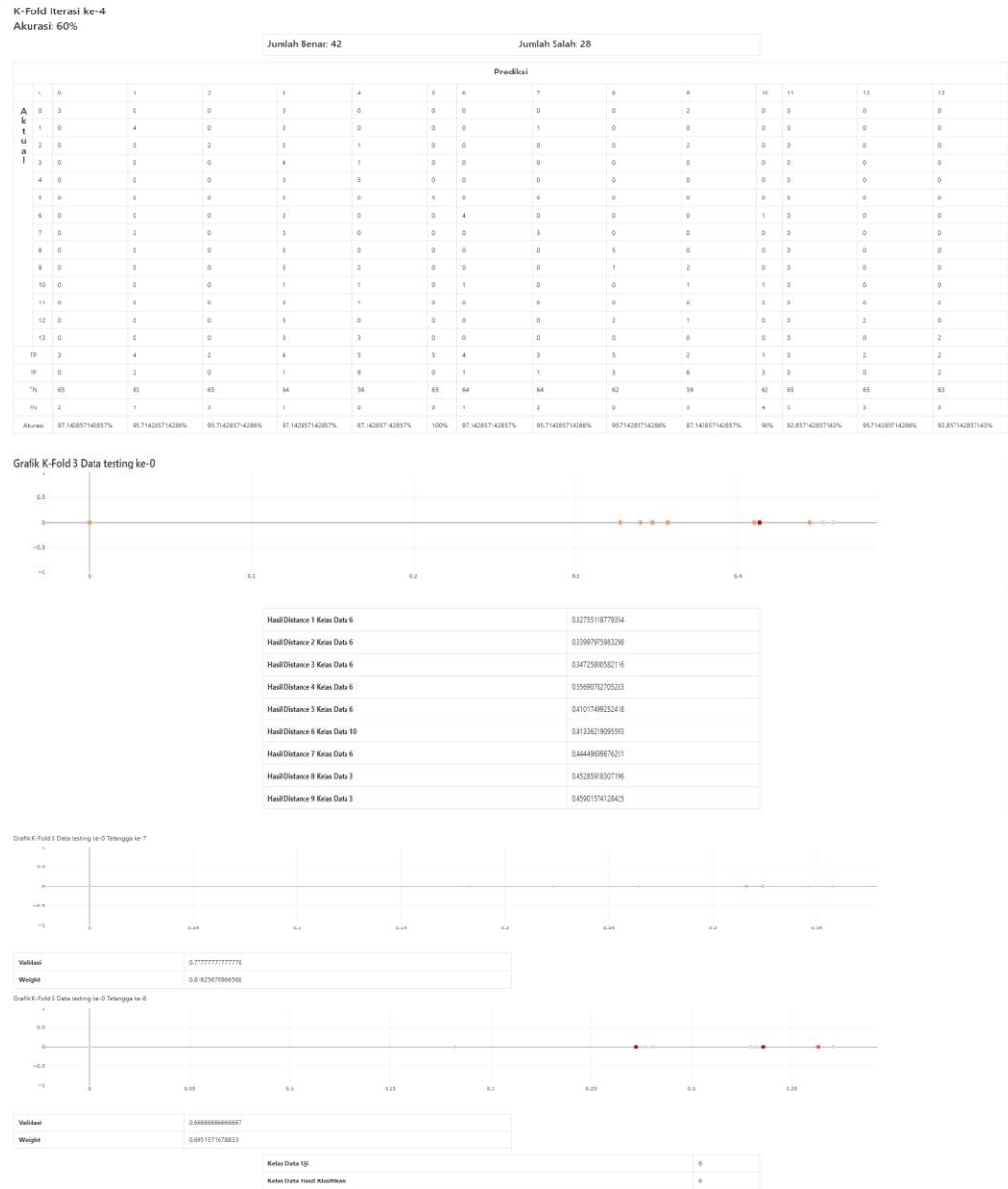
Gambar 16. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-2

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 17, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-3.



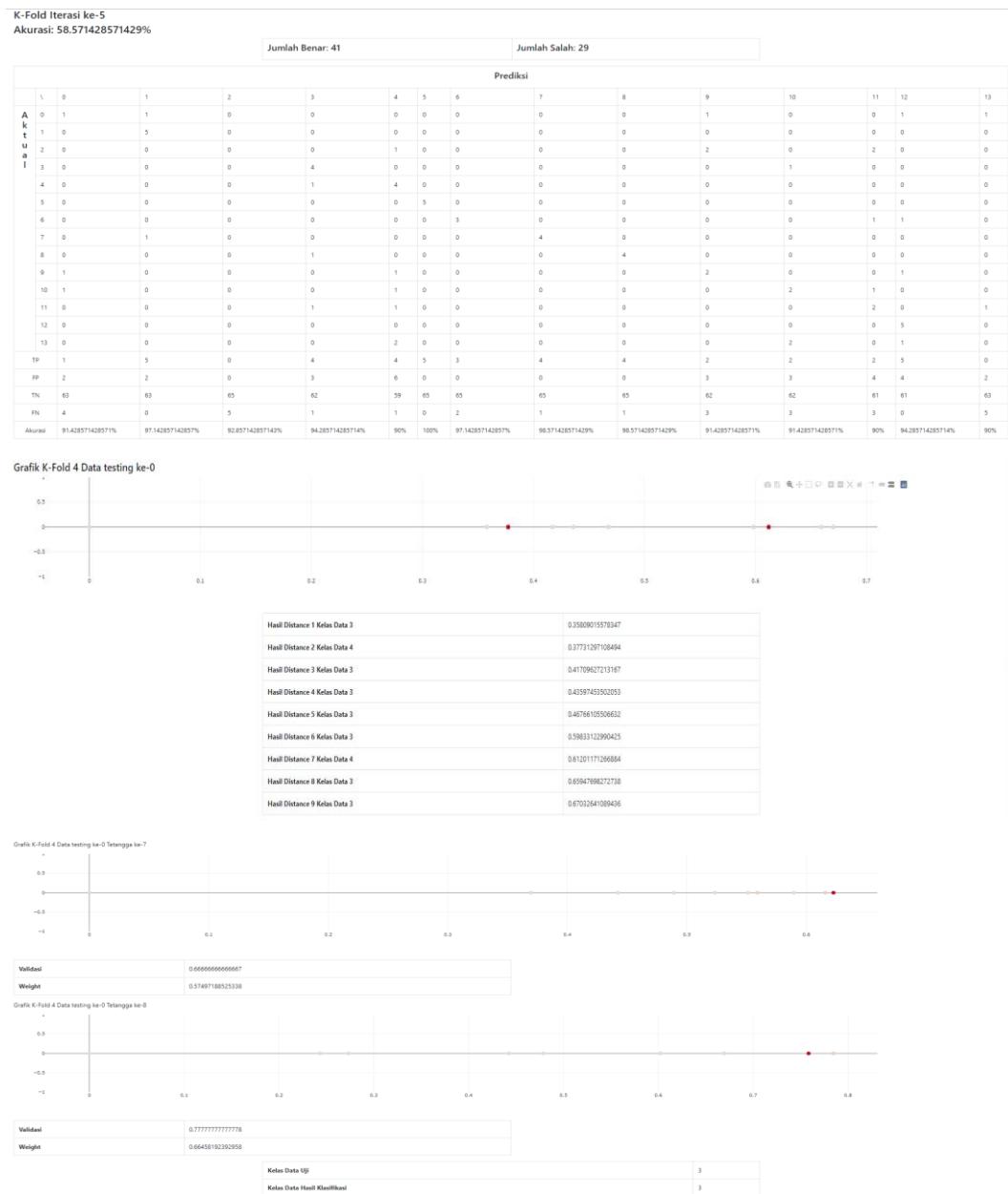
Gambar 17. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-3

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 18, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-4.



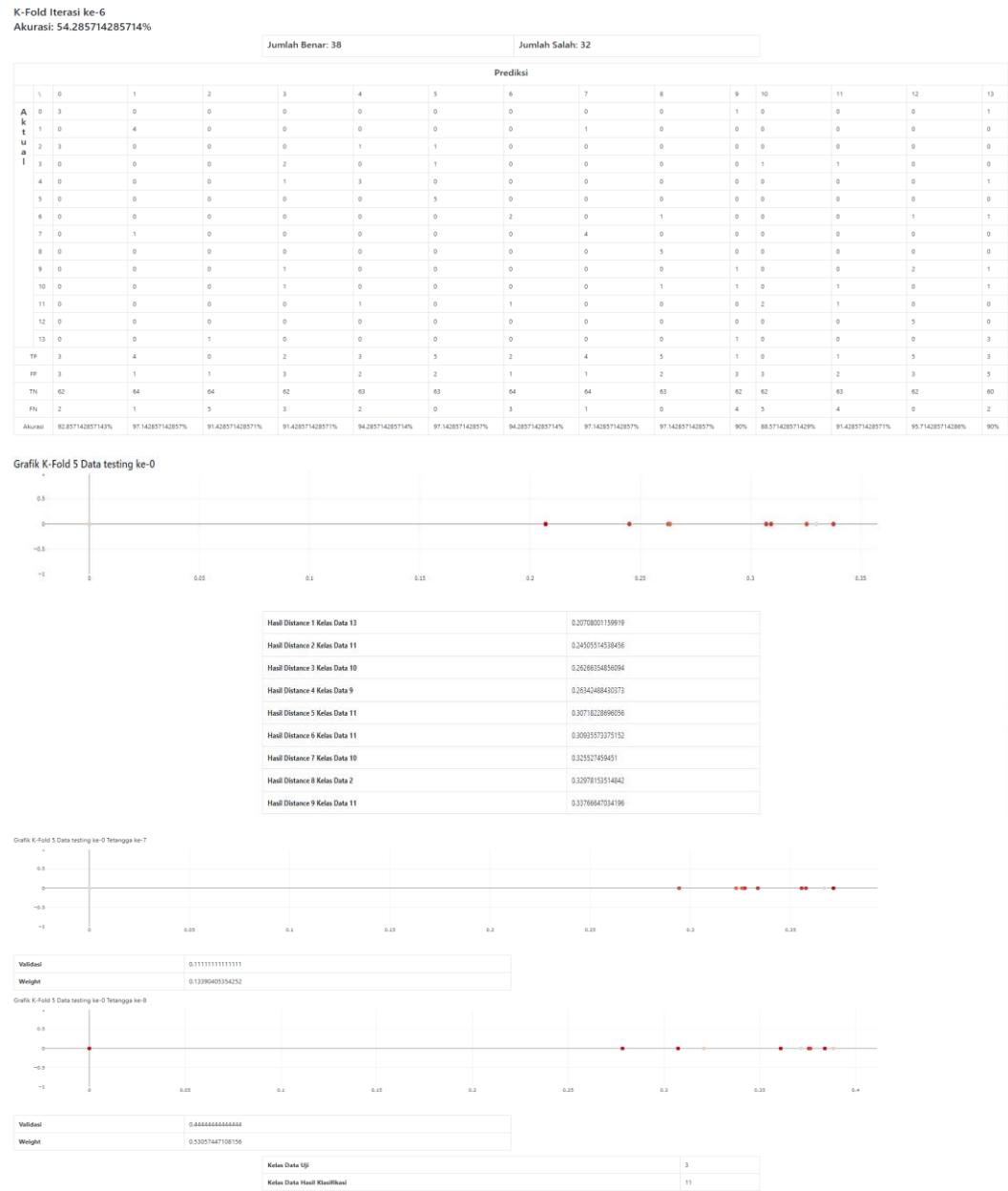
Gambar 18. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-4

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 19, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-5.



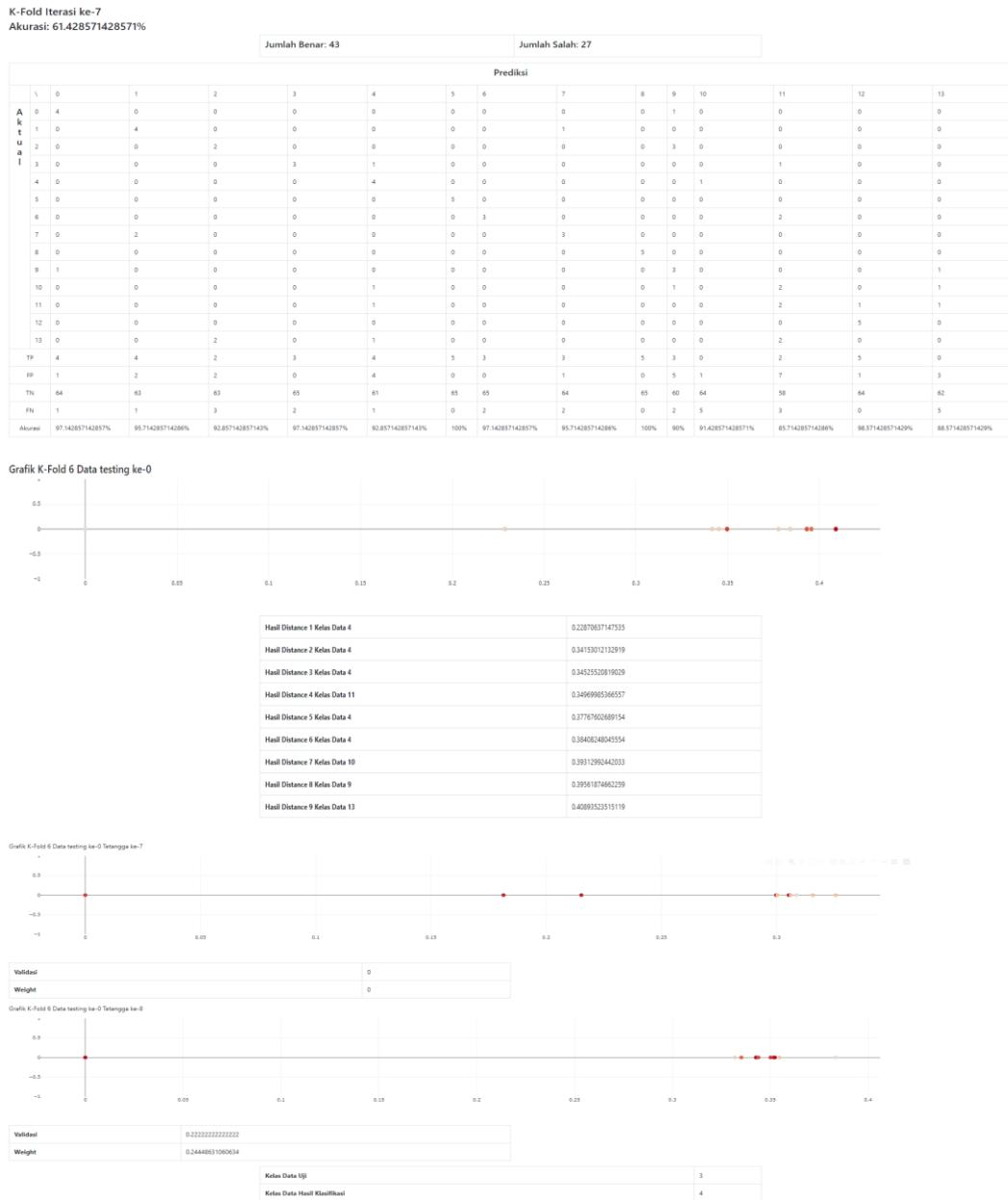
Gambar 19. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-5

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 20, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-6.



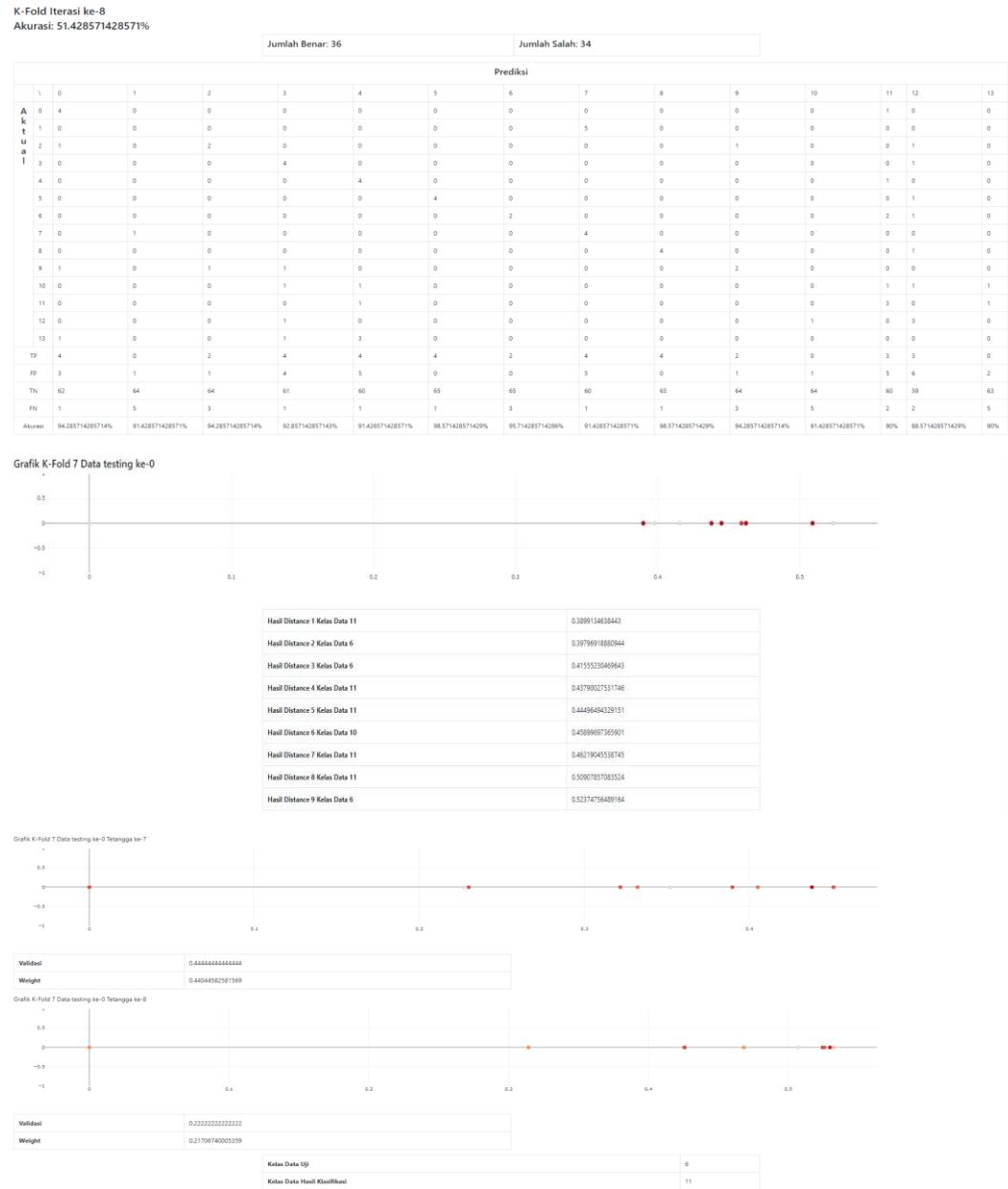
Gambar 20. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-6

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 21, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-7.



Gambar 21. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-7

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 22, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-8.



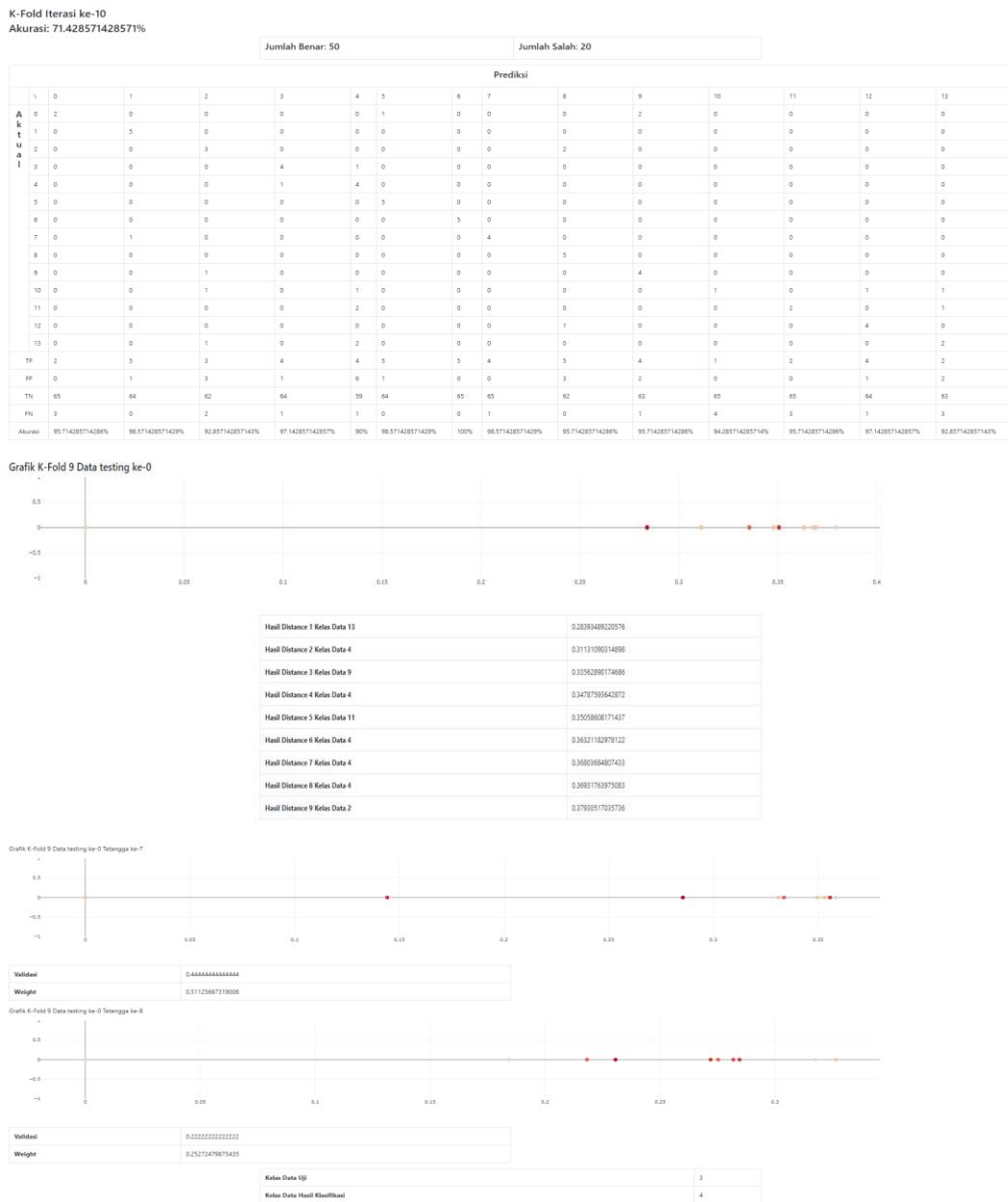
Gambar 22. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-8

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 23, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-9.



Gambar 23. Output Klasifikasi PCA–MKNN K-Fold Iterasi ke-9

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 24, merupakan hasil dari klasifikasi PCA–MKNN K-Fold Iterasi ke-10.



Gambar 24. Output Klasifikasi MKNN K-Fold Iterasi ke-10

Output dari PCA–MKNN akan munjul jika nilai  $k$  telah diinputkan. Tampilan pada Gambar 25, merupakan hasil dari rata–rata akurasi semua K-Fold klasifikasi PCA–MKNN.

Total Rata-Rata Akurasi Semua k-fold: 59.285714285714%

Gambar 25. Output Rata–Rata Akurasi Semua K-Fold Klasifikasi PCA–MKNN

## LAMPIRAN D

### Hasil Akurasi Penelitian

**Lampiran D.1. Disajikan Hasil Perbandingan Akurasi pada perubahan nilai  $k$  MKNN dengan KNN di Pengujian Pertama**

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
MKNN	56.90%	59.10%	57.70%	59.60%	57%
	k=13	k=15	K=17	k=19	k=21
	56.40%	55.10%	55.30%	54.60%	54.60%
	k=23	k=25	k=27	k=29	
	53.40%	53.10%	51.10%	50.90%	
KNN	k=3	k=5	k=7	k=9	k=11
	57.10%	60.30%	59.10%	60.70%	60.70%
	k=13	k=15	K=17	k=19	k=21
	59%	59.30%	58.70%	58.70%	58.40%
	k=23	k=25	k=27	k=29	
	57%	57%	55.90%	54.70%	

**Lampiran D.2. Disajikan Hasil Perbandingan Akurasi 10-Fold Cross-Validation MKNN dengan KNN pada  $k = 9$  di Pengujian Pertama**

10-Fold Cross-Validation	k=9 MKNN	k=9 KNN
1	65.70%	75.70%
2	48.60%	50%
3	60%	61.40%
4	60%	62.90%
5	60%	62.90%
6	57.10%	55.70%
7	62.90%	58.60%
8	54.30%	52.90%
9	55.70%	60%
10	71.40%	67.10%

**Lampiran D.3. Disajikan Hasil Evaluasi Confusion Matrix pada Klasifikasi Genre Lagu menggunakan KNN saat nilai  $k = 9$  di tiap  $k$ -fold cross-validation**

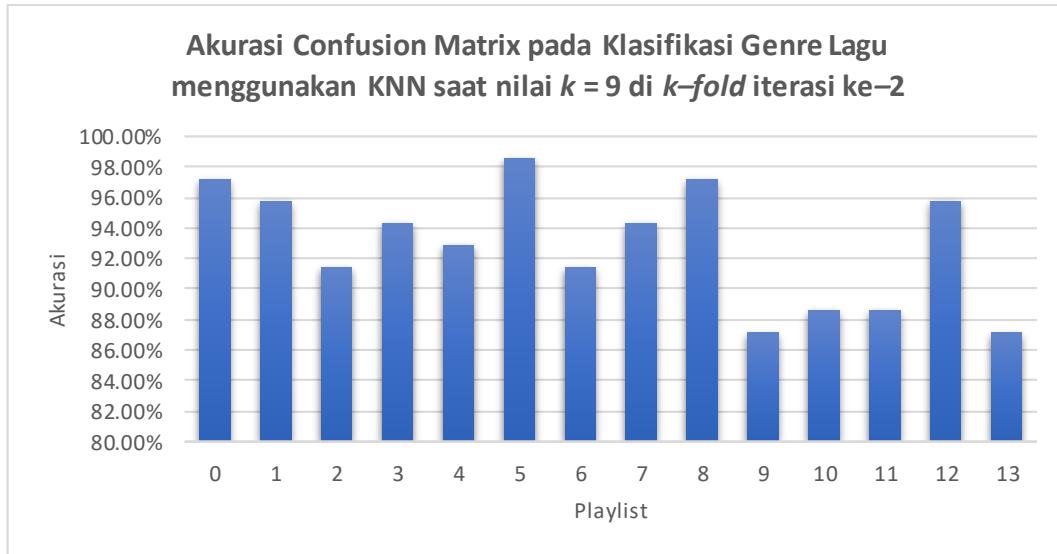
Hasil evaluasi pada klasifikasi *genre* lagu menggunakan metode KNN dengan *confusion matrix* untuk mengukur seberapa tepat klasifikasi terhadap kelas dalam

pengujian data uji dengan data latih. Proses ini dilakukan dengan *confusion matrix* menggunakan perhitungan akurasi pada Persamaan (2.7) Untuk tabel *confusion matrix* lainnya disajikan sebagai berikut:

**1. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-2.**

Tabel 1. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-2

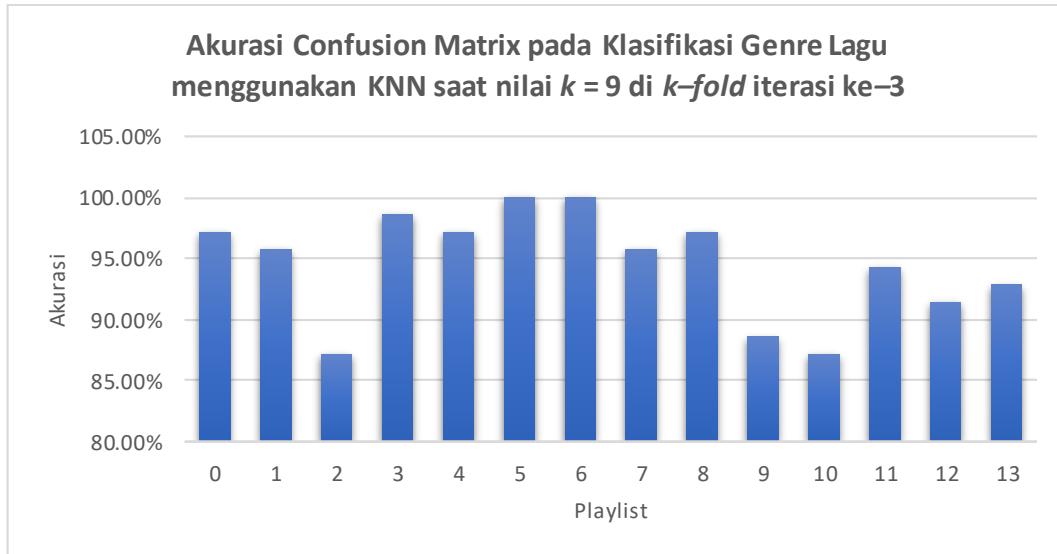
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	3	0	1	0	0	0	0	0	0	1	0	0	0	0	0
	1	0	2	0	0	0	0	0	0	3	0	0	0	0	0	0
	2	0	0	1	0	0	0	0	0	0	2	1	1	0	0	0
	3	0	0	0	3	1	0	0	1	0	0	0	0	0	0	0
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	1	1	0	0	1	0	1	1	0	0
	7	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	1	2	0	0	0	2
	10	0	0	0	1	1	0	1	0	0	1	0	1	0	0	0
	11	0	0	0	0	2	0	0	0	0	0	0	0	0	0	3
	12	0	0	0	0	0	0	0	0	2	0	0	0	0	3	0
	13	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1
TP		3	2	1	3	5	5	1	5	5	1	0	0	3	1	
FP		0	0	2	2	5	1	2	4	2	5	3	3	1	5	
TN		65	65	63	63	60	64	63	61	63	60	62	62	64	60	
FN		2	3	4	2	0	0	4	0	0	4	5	5	2	4	



**2. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-3**

Tabel 2. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-3.

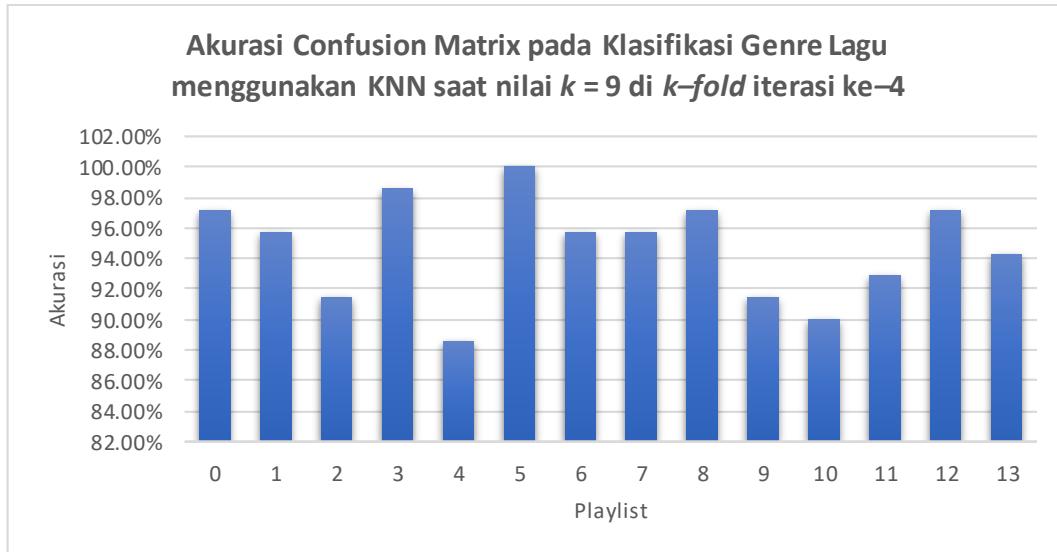
		Prediksi														
\		0	1	2	3	4	5	6	7	8	9	10	11	12	13	
A k t u a l	0	4	0	0	0	0	0	0	0	0	0	0	0	1	0	
	1	0	3	0	0	0	0	0	2	0	0	0	0	0	0	
	2	0	0	1	0	0	0	0	0	0	0	2	0	1	1	
	3	0	0	0	4	0	0	0	0	0	1	0	0	0	0	
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	
	6	0	0	0	0	0	0	5	0	0	0	0	0	0	0	
	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	
	9	0	0	3	0	0	0	0	0	0	0	2	0	0	0	
	10	0	0	1	0	1	0	0	0	0	0	1	0	2	0	
	11	0	0	1	0	0	0	0	0	0	1	1	1	0	1	
	12	0	0	0	0	0	0	0	0	2	0	0	0	3	0	
	13	1	0	0	0	1	0	0	0	0	1	0	0	0	2	
TP		4	3	1	4	5	5	5	4	5	0	1	1	3	2	
FP		1	1	5	0	2	0	0	2	2	3	5	0	4	2	
TN		64	64	60	65	63	65	65	63	63	62	60	65	61	63	
FN		1	2	4	1	0	0	0	1	0	5	4	4	2	3	



### 3. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-4

Tabel 3. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-4.

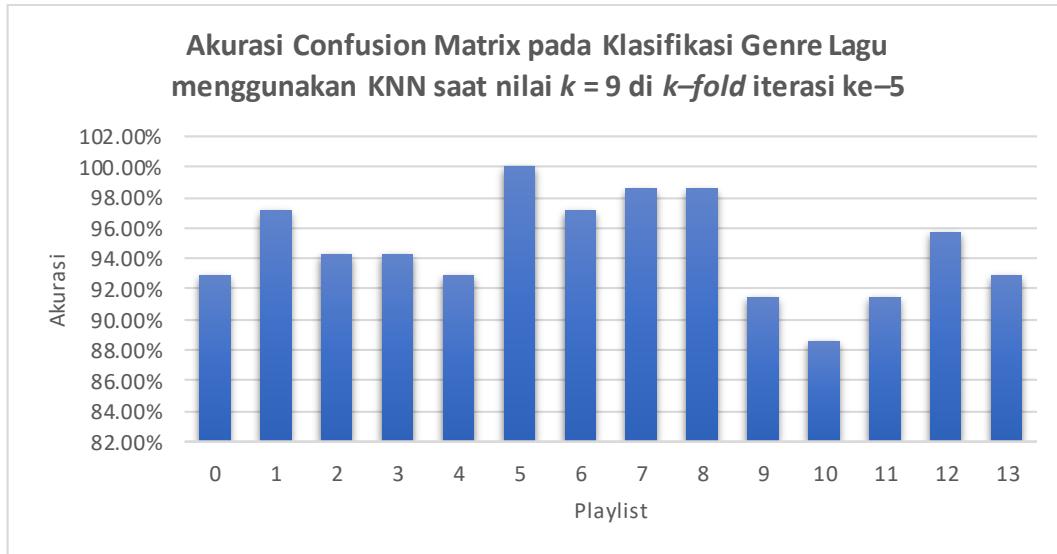
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	3	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
	2	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0
	3	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	1	0	0	0	3	0	0	0	1	0	0	0	0
	7	0	2	0	0	0	0	0	0	3	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	9	0	0	1	0	1	0	0	0	0	0	2	0	0	0	1
	10	0	0	1	0	1	0	1	0	0	1	1	1	0	0	0
	11	0	0	0	0	2	0	0	0	0	0	2	0	0	1	0
	12	0	0	0	0	0	0	0	0	2	0	0	0	0	3	0
	13	0	0	0	0	2	0	0	0	0	0	0	0	0	0	3
TP		3	4	3	4	5	5	3	3	5	2	1	0	3	3	
FP		0	2	4	0	8	0	1	1	2	3	3	0	0	2	
TN		65	63	61	65	57	65	64	64	63	62	62	65	65	63	
FN		2	1	2	1	0	0	2	2	0	3	4	5	2	2	



**4. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-5.**

Tabel 4. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-5

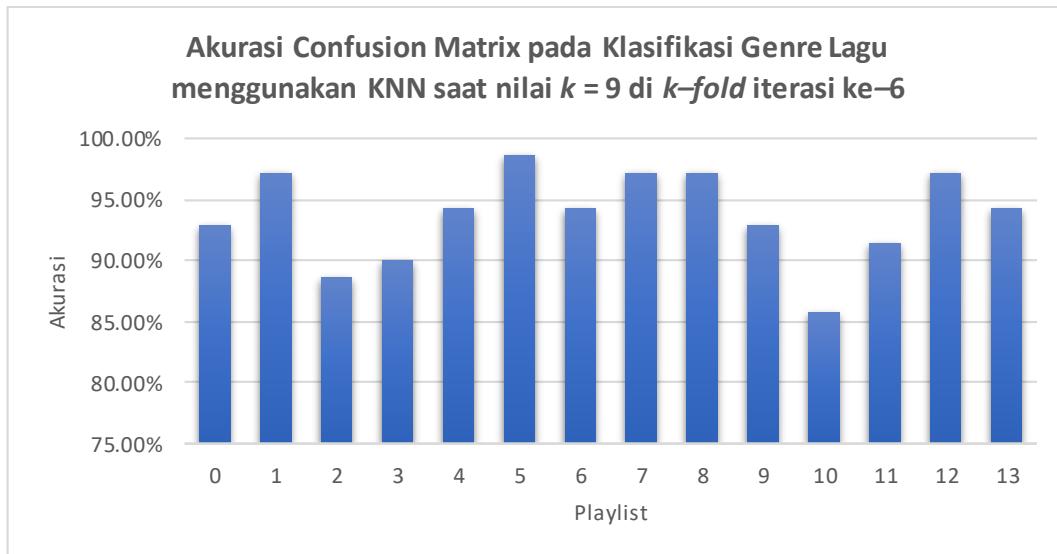
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	2	1	0	0	0	0	0	0	0	0	0	0	0	1	1
	1	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	2	0	0	0	0	0	0	0	1	1	1	0	0
	3	0	0	0	4	0	0	0	0	0	0	0	1	0	0	0
	4	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	3	0	0	0	0	0	1	1	0
	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0
	9	1	0	1	0	0	0	0	0	0	2	0	0	0	1	0
	10	1	0	0	0	1	0	0	0	0	1	1	1	0	0	0
	11	0	0	0	1	1	0	0	0	0	1	0	2	0	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	13	0	0	0	0	2	0	0	0	0	0	2	0	0	0	1
TP		2	5	2	4	4	5	3	4	4	2	1	2	5	1	
FP		2	2	1	3	4	0	0	0	0	3	4	3	3	1	
TN		63	63	64	62	61	65	65	65	65	62	61	62	62	64	
FN		3	0	3	1	1	0	2	1	1	3	4	3	0	4	



##### 5. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-6

Tabel 5. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-6.

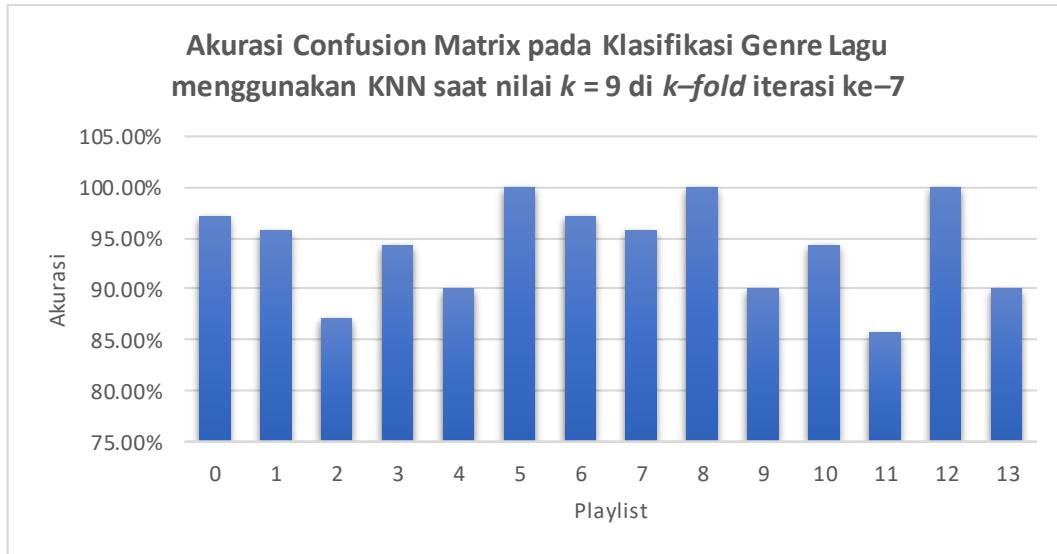
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
	2	3	0	0	0	0	1	0	0	0	0	0	1	0	0	0
	3	0	0	0	2	1	0	0	0	0	0	0	1	1	0	0
	4	0	0	0	1	3	0	0	0	0	0	0	0	0	0	1
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	2	0	1	0	2	0	0	0	0
	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	0	0	0	1	0	0	0	0	0	2	0	0	0	2	0
	10	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0
	11	0	0	0	1	1	0	1	0	0	0	1	1	1	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	13	0	0	1	0	0	0	0	0	0	0	1	1	0	0	2
TP		3	4	0	2	3	5	2	4	5	2	1	1	5	2	
FP		3	1	3	4	2	1	1	1	2	2	6	2	2	1	
TN		62	64	62	61	63	64	64	64	63	63	59	63	63	64	
FN		2	1	5	3	2	0	3	1	0	3	4	4	0	3	



**6. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-7.**

Tabel 6. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-7

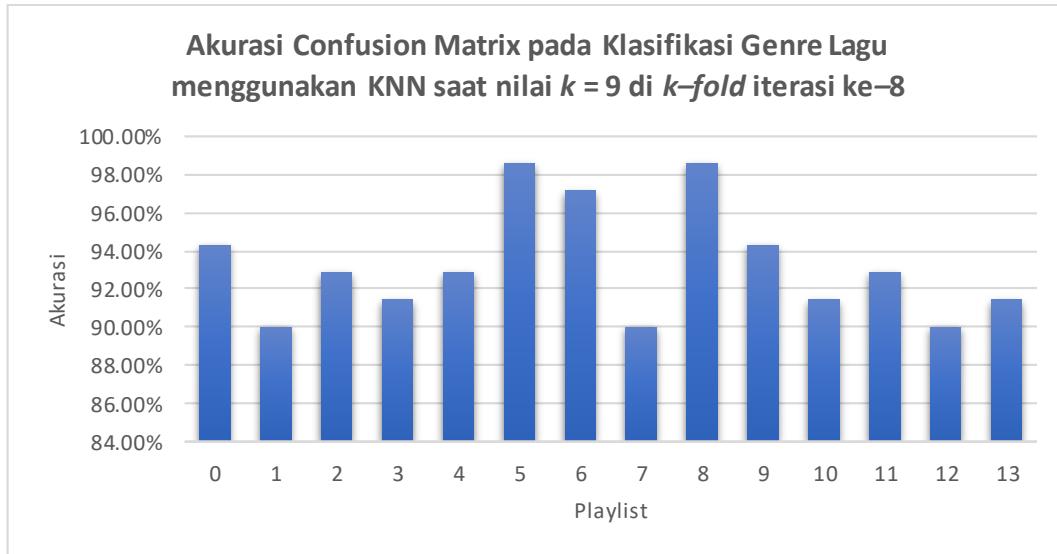
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
	2	0	0	2	0	0	0	0	0	0	0	3	0	0	0	0
	3	0	0	1	2	1	0	0	0	0	0	0	0	1	0	0
	4	0	0	1	1	3	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	3	0	0	0	0	0	2	0	0
	7	0	2	0	0	0	0	0	0	3	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	9	1	0	0	0	0	0	0	0	0	0	3	0	0	0	1
	10	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1
	11	0	0	1	0	2	0	0	0	0	1	0	1	1	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	13	0	0	2	0	1	0	0	0	0	0	0	0	2	0	0
TP		4	4	2	2	3	5	3	3	5	3	1	1	5	0	0
FP		1	2	6	1	5	0	0	1	0	5	0	6	0	2	0
TN		64	63	59	64	60	65	65	64	65	60	65	59	65	63	0
FN		1	1	3	3	2	0	2	2	0	2	4	4	0	5	0



**7. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k\text{-fold}$  iterasi ke-8.**

Tabel 7. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k\text{-fold}$  iterasi ke-8

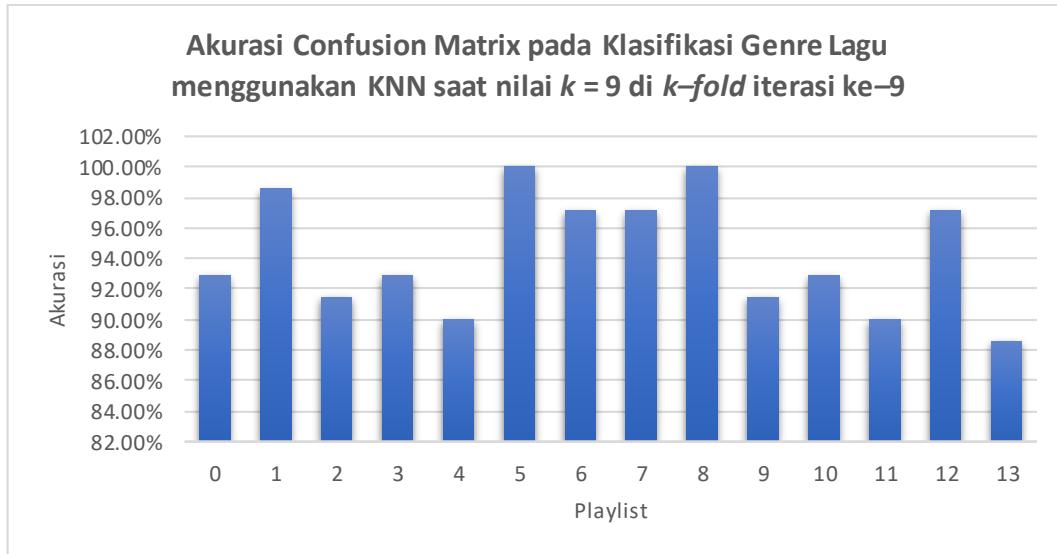
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	4	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	1	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0
	2	1	0	2	0	0	0	0	0	0	0	1	0	0	1	0
	3	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0
	4	0	0	0	0	4	0	0	0	0	0	0	1	0	0	0
	5	0	0	0	0	0	4	0	0	0	0	0	0	0	1	0
	6	0	0	0	0	0	0	3	0	0	0	0	0	1	1	0
	7	0	3	0	0	0	0	0	0	2	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0
	9	1	0	1	1	0	0	0	0	0	0	2	0	0	0	0
	10	0	0	0	2	0	0	0	0	0	0	0	2	0	0	1
	11	0	0	1	0	1	0	0	0	0	0	0	1	2	0	0
	12	0	0	0	1	0	0	0	0	0	0	0	1	0	3	0
	13	1	0	0	1	3	0	0	0	0	0	0	0	0	0	0
TP		4	1	2	4	4	4	3	2	4	2	2	2	3	0	
FP		3	3	2	5	4	0	0	4	0	1	3	2	5	1	
TN		62	62	63	60	61	65	65	61	65	64	62	63	60	64	
FN		1	4	3	1	1	1	2	3	1	3	3	3	2	5	



**8. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-9.**

Tabel 8. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-9

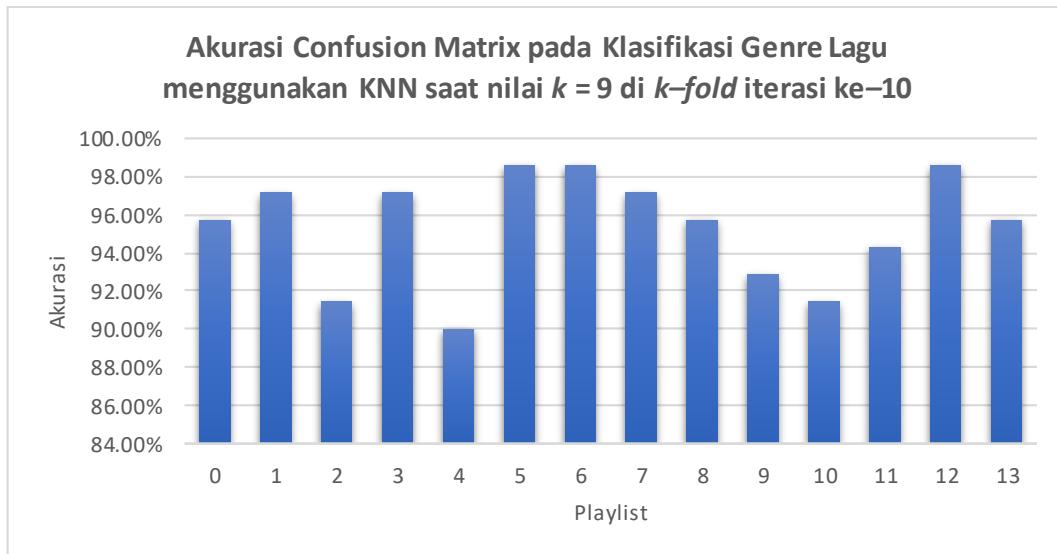
		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0
k	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
t	2	2	0	1	0	1	0	0	0	0	0	1	0	0	0	0
u	3	0	0	0	3	1	0	0	0	0	0	0	0	0	0	1
a	4	0	0	1	0	2	0	0	0	0	0	0	0	0	0	2
l	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	1	0	0	4	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	2	0	0	0	0	0	0	0	0	0	2	0	0	1	0
	10	0	0	0	0	1	0	1	0	0	2	1	0	0	0	0
	11	0	0	1	0	1	0	0	0	0	0	1	0	0	2	0
	12	0	0	0	1	0	0	0	0	0	0	0	0	0	4	0
	13	0	0	0	1	0	0	0	0	0	0	0	2	0	0	2
TP		4	4	1	3	2	5	4	5	5	2	1	0	4	2	
FP		4	0	2	3	4	0	1	2	0	3	1	2	1	5	
TN		61	65	63	62	61	65	64	63	65	62	64	63	64	60	
FN		1	1	4	2	3	0	1	0	0	3	4	5	1	3	



**9. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k\text{-fold}$  iterasi ke-10.**

Tabel 9. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan KNN saat nilai  $k = 9$  di  $k\text{-fold}$  iterasi ke-10

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	2	0	0	0	0	0	1	0	0	0	2	0	0	0	0
k	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
t	2	0	0	3	0	0	0	0	0	0	2	0	0	0	0	0
u	3	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0
a	4	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0
l	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	4	0	0	0	0	1	0	0	0
	7	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	9	0	0	2	0	0	0	0	0	0	0	3	0	0	0	0
	10	0	0	2	0	0	0	0	0	0	0	1	1	1	0	0
	11	0	0	0	0	2	0	0	0	0	0	0	1	2	0	0
	12	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0
	13	0	0	0	0	3	0	0	0	0	0	0	0	0	0	2
TP		2	4	3	4	4	5	4	4	5	3	1	2	4	2	
FP		0	1	4	1	6	1	0	1	3	3	2	1	0	0	
TN		65	64	61	64	59	64	65	64	62	62	63	64	65	65	
FN		3	1	2	1	1	0	1	1	0	2	4	3	1	3	



**Lampiran D.4. Disajikan Hasil Evaluasi *Confusion Matrix* pada Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di tiap  $k$ -fold cross-validation**

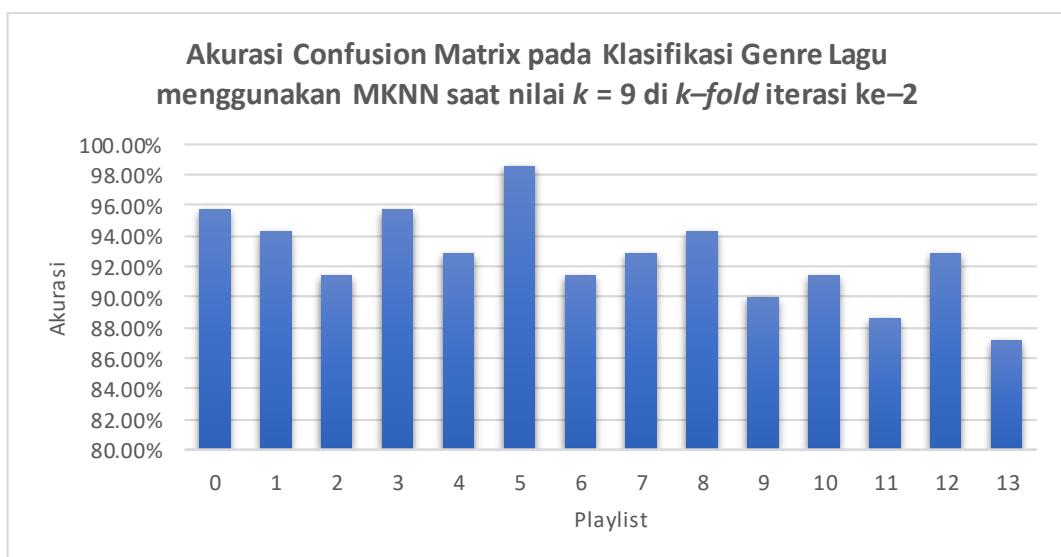
Hasil evaluasi pada klasifikasi *genre lagu* menggunakan metode MKNN dengan *confusion matrix* untuk mengukur seberapa tepat klasifikasi terhadap kelas dalam pengujian data uji dengan data latih. Proses ini dilakukan dengan *confusion matrix* menggunakan perhitungan akurasi pada Persamaan (2.7) Untuk tabel *confusion matrix* lainnya disajikan sebagai berikut:

**1. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-2.**

Tabel 1. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-2

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	3	0	1	0	0	0	0	0	0	1	0	0	0	0	
	1	0	1	0	0	0	0	0	0	4	0	0	0	0	0	
	2	0	0	1	0	0	0	0	0	0	0	1	1	1	0	
	3	0	0	0	3	1	0	0	1	0	0	0	0	0	0	
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	
	6	0	0	0	0	0	1	1	0	1	0	0	1	1	0	

	7	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	4	0	0	0	1	0
	9	0	0	0	1	0	0	0	0	0	1	0	0	0	3
	10	0	0	1	0	1	0	1	0	0	1	0	1	0	0
	11	0	0	0	0	2	0	0	0	0	0	0	0	0	3
	12	0	0	0	0	0	0	0	0	2	0	0	0	3	0
	13	1	0	0	0	1	0	1	0	0	0	0	0	0	2
TP	3	1	1	3	5	5	1	5	4	1	0	0	3	2	
FP	1	0	2	1	5	1	2	5	3	3	1	3	3	6	
TN	64	65	63	64	60	64	63	60	62	62	64	62	62	59	
FN	2	4	4	2	0	0	4	0	1	4	5	5	2	3	

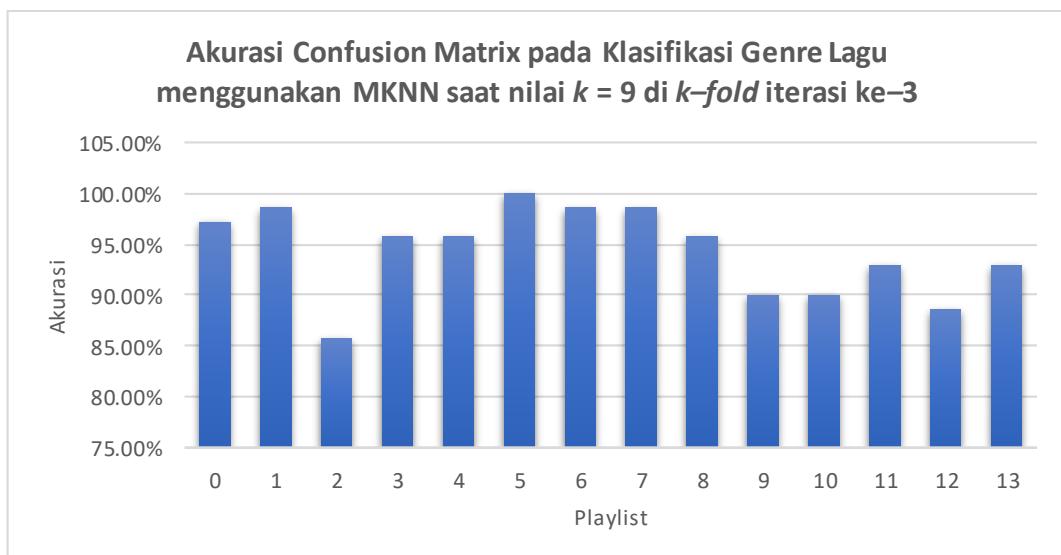


2. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-3.

Tabel 2. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-3

Prediksi															
	\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	4	0	0	0	0	0	0	0	0	0	0	0	1	0
	1	0	4	0	0	0	0	0	1	0	0	0	0	0	0
	2	0	0	1	0	0	0	0	0	0	0	1	1	1	1
	3	0	0	0	4	0	0	0	0	0	1	0	0	0	0
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	4	0	0	0	0	0	1	0

	7	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	0	0	3	1	0	0	0	0	0	0	1	0	0	0	0
	10	0	0	1	0	1	0	0	0	0	0	1	0	2	0	0
	11	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0
	12	0	0	0	0	0	0	0	0	3	0	0	0	2	0	0
	13	1	0	1	0	1	0	0	0	0	1	0	0	0	1	0
TP	4	4	1	4	5	5	4	5	5	0	1	1	2	1		
FP	1	0	6	2	3	0	0	1	3	2	3	1	5	1		
TN	64	65	59	63	62	65	65	64	62	63	62	64	60	64		
FN	1	1	4	1	0	0	1	0	0	5	4	4	3	4		

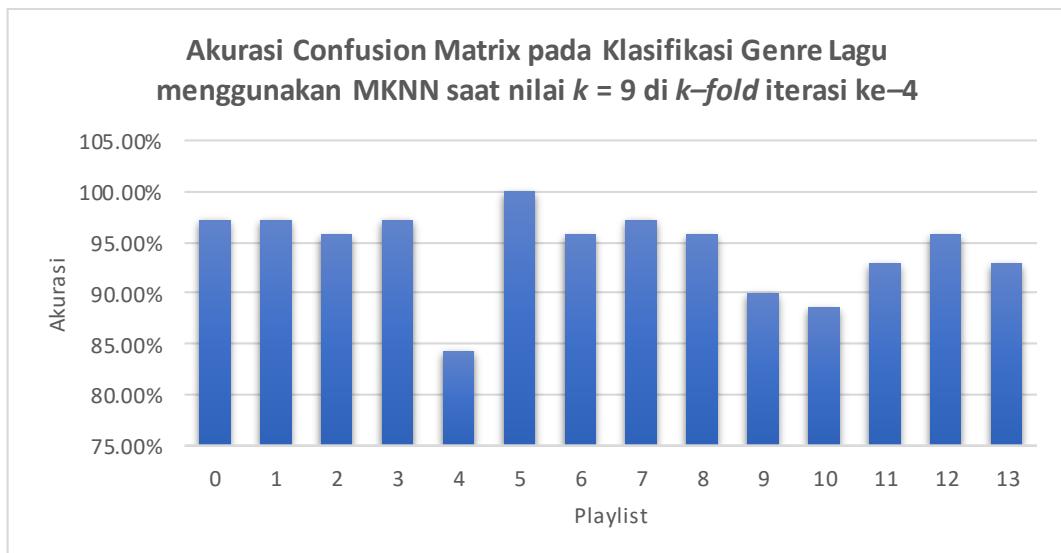


### 3. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-4.

Tabel 3. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-4

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	3	0	0	0	0	0	0	0	0	2	0	0	0	0	0
	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
	2	0	0	2	0	1	0	0	0	0	2	0	0	0	0	0
	3	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	3	0	0	0	1	0	0	0	1

	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	0	0	0	0	2	0	0	0	1	2	0	0	0	0	0
	10	0	0	0	1	2	0	1	0	0	0	0	0	1	0	0
	11	0	0	0	0	2	0	0	0	0	0	2	0	0	1	0
	12	0	0	0	0	0	0	0	0	0	2	0	0	0	3	0
	13	0	0	0	0	3	0	0	0	0	0	0	0	0	2	0
TP	3	4	2	4	5	5	3	4	5	2	0	0	3	2		
FP	0	1	0	1	11	0	1	1	3	4	3	0	1	2		
TN	65	64	65	64	54	65	64	64	62	61	62	65	64	63		
FN	2	1	3	1	0	0	2	1	0	3	5	5	2	3		



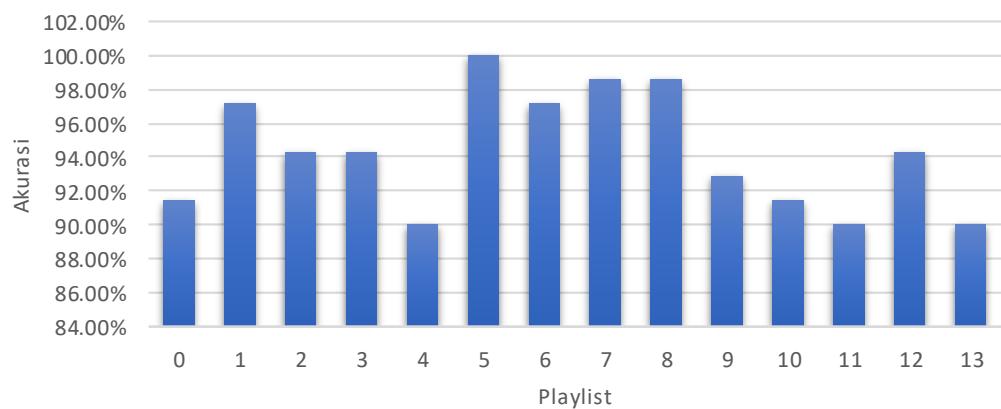
#### 4. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-5.

Tabel 4. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-5

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	1
	1	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	1	0	1	0	0	0	0	0	1	0	2	0	0
	3	0	0	0	4	0	0	0	0	0	0	0	1	0	0	0
	4	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	3	0	0	0	0	1	1	0	

	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0
	9	1	0	0	0	1	0	0	0	0	2	0	0	1	0	
	10	1	0	0	0	1	0	0	0	0	0	2	1	0	0	
	11	0	0	0	1	1	0	0	0	0	0	0	2	0	1	
	12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
	13	0	0	0	0	2	0	0	0	0	0	2	0	1	0	
TP	1	5	1	4	4	5	3	4	4	2	2	2	5	0		
FP	2	2	0	3	6	0	0	0	0	2	3	4	4	2		
TN	63	63	65	62	59	65	65	65	65	63	62	61	61	63		
FN	4	0	4	1	1	0	2	1	1	3	3	3	0	5		

**Akurasi Confusion Matrix pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-5**



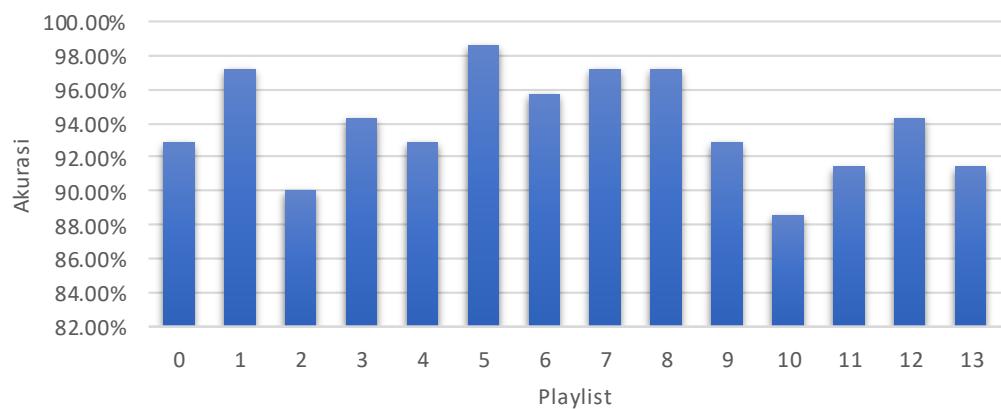
##### 5. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-6.

Tabel 5. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-6

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	3	0	0	0	0	0	0	0	0	0	1	0	0	0	1
	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
	2	3	0	0	0	1	1	0	0	0	0	0	0	0	0	0
	3	0	0	0	2	1	0	0	0	0	0	0	1	1	0	0
	4	0	0	0	1	3	0	0	0	0	0	0	0	0	0	1
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	3	0	1	0	0	0	1	0	0

	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	9	0	0	1	0	0	0	0	0	0	2	0	0	2	0	0
	10	0	0	0	0	0	0	0	0	1	0	0	1	1	2	0
	11	0	0	0	0	1	0	1	0	0	0	2	1	0	0	0
	12	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	13	0	0	1	0	0	0	0	0	0	1	0	0	0	3	0
TP	3	4	0	2	3	5	3	4	5	2	0	1	5	3		
FP	3	1	2	1	3	1	1	1	2	2	3	2	4	4		
TN	62	64	63	64	62	64	64	64	63	63	62	63	61	61		
FN	2	1	5	3	2	0	2	1	0	3	5	4	0	2		

**Akurasi Confusion Matrix pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-6**

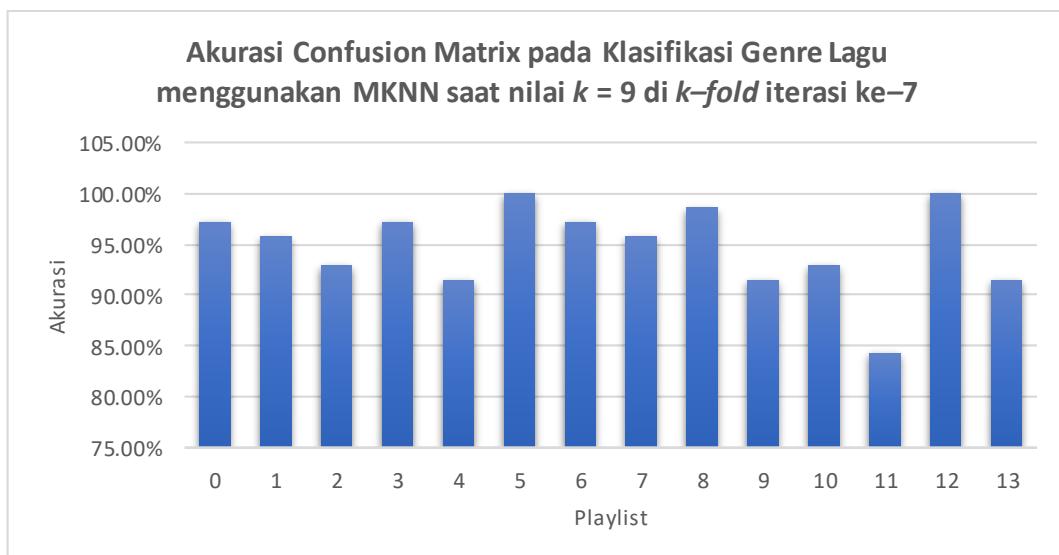


#### 6. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-7.

Tabel 6. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-7

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
	2	0	0	1	0	0	0	0	0	0	1	3	0	0	0	0
	3	0	0	0	3	1	0	0	0	0	0	0	0	1	0	0
	4	0	0	0	0	4	0	0	0	0	0	0	0	1	0	0
	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	3	0	0	0	0	2	0	0	0

	7	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	9	1	0	0	0	0	0	0	0	4	0	0	0	0	0	0
	10	0	0	0	0	1	0	0	0	1	0	2	0	1		
	11	0	0	0	0	2	0	0	0	0	0	2	0	1		
	12	0	0	0	0	0	0	0	0	0	0	0	5	0		
	13	0	0	1	0	1	0	0	0	0	0	2	0	1		
TP	4	4	1	3	4	5	3	3	5	4	0	2	5	1		
FP	1	2	1	0	5	0	0	1	1	5	0	8	0	2		
TN	64	63	64	65	60	65	65	64	64	60	65	57	65	63		
FN	1	1	4	2	1	0	2	2	0	1	5	3	0	4		



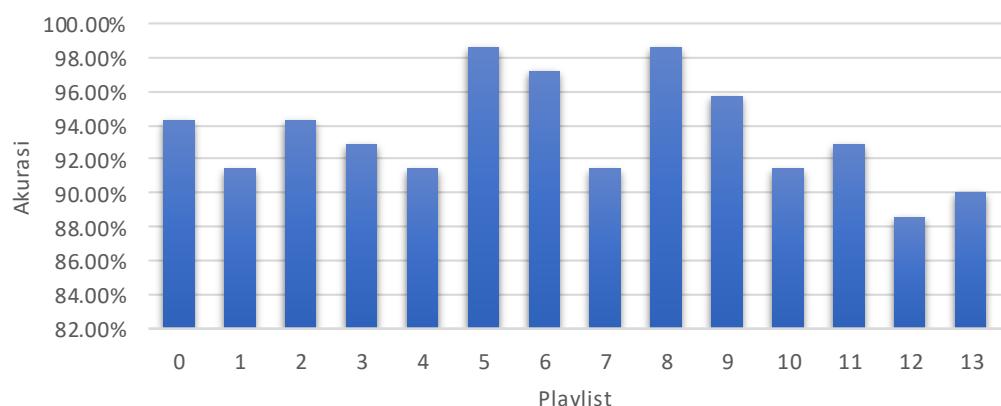
#### 7. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-8.

Tabel 7. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-8

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A k t u a l	0	4	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	1	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	2	1	0	2	0	0	0	0	0	0	0	1	0	0	1	0
	3	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0
	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1
	5	0	0	0	0	0	4	0	0	0	0	0	0	0	1	0
	6	0	0	0	0	0	0	3	0	0	0	0	0	1	1	0

	7	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	4	0	0	0	0	1	0
	9	1	0	0	1	0	0	0	0	0	3	0	0	0	0	0
	10	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1
	11	0	0	1	0	1	0	0	0	0	0	0	3	0	0	0
	12	0	0	0	1	0	0	0	0	0	0	1	0	3	0	0
	13	1	0	0	1	3	0	0	0	0	0	0	0	0	0	0
TP	4	0	2	4	4	4	3	4	4	3	0	3	3	3	0	
FP	3	1	1	4	5	0	0	5	0	1	1	3	6	2		
TN	62	64	64	61	60	65	65	60	65	64	64	62	59	63		
FN	1	5	3	1	1	1	2	1	1	2	5	2	2	5		

**Akurasi Confusion Matrix pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-8**



#### 8. Hasil Evaluasi Confusion Matrix Pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai $k = 9$ di $k$ -fold iterasi ke-9.

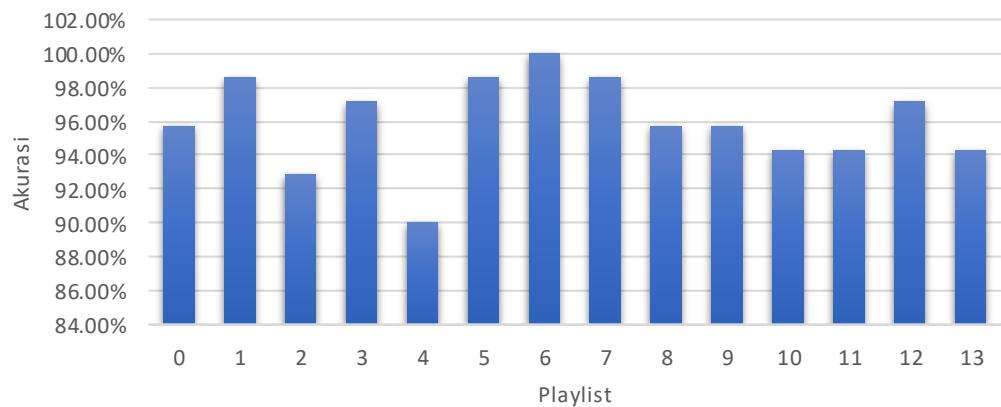
Tabel 8. Hasil evaluasi *Confusion Matrix* Klasifikasi *Genre Lagu* menggunakan MKNN saat nilai  $k = 9$  di  $k$ -fold iterasi ke-9

		Prediksi														
		\	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0
k	1	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
t	2	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1
u	3	0	0	0	2	1	0	0	0	0	0	0	0	0	1	1
a	4	0	0	0	0	2	0	0	0	0	1	0	0	0	0	2
l	5	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
	6	0	0	0	1	0	0	3	0	0	0	0	0	1	0	0



7	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0
10	0	0	1	0	1	0	0	0	0	0	1	1	1	0	1	0
11	0	0	0	0	2	0	0	0	0	0	0	2	0	1	0	1
12	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0	0
13	0	0	0	0	2	0	0	0	0	0	0	0	0	0	3	0
TP	2	5	2	4	4	5	5	4	5	4	1	2	4	3		
FP	0	1	2	1	6	1	0	0	3	2	0	1	1	2		
TN	65	64	63	64	59	64	65	65	62	63	65	64	64	63		
FN	3	0	3	1	1	0	0	1	0	1	4	3	1	2		

**Akurasi Confusion Matrix pada Klasifikasi Genre Lagu menggunakan MKNN saat nilai  $k = 9$  di  $k\text{-fold}$  iterasi ke-10**



**Lampiran D.5. Disajikan Hasil Perbandingan Akurasi pada perubahan nilai  $k$  PCA–MKNN dengan PCA–KNN di Pengujian Kedua**

**1. Metode PCA–MKNN dengan PCA–KNN pada 1 fitur baru.**

Metode	Nilai $k$				
	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$
PCA–MKNN	17.10%	18.40%	19.70%	20.60%	23.30%
	$k=13$	$k=15$	K=17	$k=19$	$k=21$
	23.10%	23.10%	23%	23.10%	23%
	$k=23$	$k=25$	$k=27$	$k=29$	
	22.60%	23.90%	23.90%	24%	
PCA–KNN	$k=3$	$k=5$	$k=7$	$k=9$	$k=11$
	18.70%	21%	22.10%	20.30%	21.60%

	k=13	k=15	K=17	k=19	k=21
	22.40%	22.90%	22.90%	22.10%	23.40%
	k=23	k=25	k=27	k=29	
	22.90%	24.40%	24.30%	25.30%	

## 2. Metode PCA–MKNN dengan PCA–KNN pada 2 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA–MKNN	21.30%	24.10%	25.40%	24.70%	23.90%
	k=13	k=15	K=17	k=19	k=21
	24.90%	25.70%	24.10%	23.40%	23.90%
	k=23	k=25	k=27	k=29	
	24%	24.70%	24.70%	24.60%	
	23.30%	24.60%	23.60%	23.60%	23%
PCA–KNN	23.30%	24.60%	23.60%	23.60%	23%
	k=13	k=15	K=17	k=19	k=21
	23.40%	22.90%	22.10%	22.40%	23%
	k=23	k=25	k=27	k=29	
	23%	22.90%	24.30%	24.10%	

## 3. Metode PCA–MKNN dengan PCA–KNN pada 3 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA–MKNN	33.30%	35.30%	36%	35.40%	34.90%
	k=13	k=15	K=17	k=19	k=21
	35.10%	35.60%	34.10%	34.10%	34.70%
	k=23	k=25	k=27	k=29	
	33.90%	33.90%	34.10%	33.60%	
	32.90%	34.40%	32.90%	34.40%	33.10%
PCA–KNN	32.90%	34.40%	32.90%	34.40%	33.10%
	k=13	k=15	K=17	k=19	k=21
	33.70%	35.10%	34.40%	32.60%	33.70%
	k=23	k=25	k=27	k=29	
	33.30%	33.60%	33.90%	34.10%	

## 4. Metode PCA–MKNN dengan PCA–KNN pada 4 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA–MKNN	36.40%	40.40%	40%	39.60%	39.60%

	k=13	k=15	K=17	k=19	k=21
PCA-KNN	39.70%	37.40%	37.10%	36.70%	36.40%
	k=23	k=25	k=27	k=29	
	35.10%	35%	34.70%	34.70%	
	k=3	<b>k=5</b>	k=7	k=9	k=11
	34.90%	<b>38.90%</b>	39%	38.40%	38.90%
	k=13	k=15	K=17	k=19	k=21
38.70%	37.40%	37%	37.10%	37.30%	
k=23	k=25	k=27	k=29		
36.70%	36.40%	37%	35.40%		

##### 5. Metode PCA-MKNN dengan PCA-KNN pada 5 fitur baru.

Metode	Nilai k				
PCA-MKNN	k=3	k=5	<b>k=7</b>	k=9	k=11
	46.40%	46.70%	<b>48%</b>	45.90%	45.70%
	k=13	k=15	K=17	k=19	k=21
	45.30%	45.70%	44.40%	43.10%	43.10%
	k=23	k=25	k=27	k=29	
	42.90%	42.90%	43.10%	42.40%	
PCA-KNN	k=3	k=5	<b>k=7</b>	k=9	k=11
	45.40%	46.30%	<b>46.60%</b>	45.60%	46.70%
	k=13	k=15	K=17	k=19	k=21
	46.60%	45.10%	45.70%	44.70%	44%
	k=23	k=25	k=27	k=29	
	43.60%	44.30%	45.30%	44%	

##### 6. Metode PCA-MKNN dengan PCA-KNN pada 6 fitur baru.

Metode	Nilai k				
PCA-MKNN	k=3	k=5	k=7	k=9	k=11
	45.90%	48.70%	49.10%	48.90%	47.40%
	<b>k=13</b>	k=15	K=17	k=19	k=21
	<b>47.10%</b>	47.40%	46.60%	46.70%	46.10%
	k=23	k=25	k=27	k=29	
	46%	44.10%	43.90%	44%	
PCA-KNN	k=3	k=5	k=7	k=9	k=11
	45.10%	46.90%	49%	48.60%	47.70%
	<b>k=13</b>	k=15	K=17	k=19	k=21
	<b>49.40%</b>	48.90%	47.40%	47.60%	47.10%
	k=23	k=25	k=27	k=29	
	46.70%	45.90%	45.30%	45.10%	

### 7. Metode PCA–MKNN dengan PCA–KNN pada 7 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA–MKNN	48.10%	49.60%	51.90%	51.10%	49.30%
	k=13	k=15	K=17	k=19	k=21
	47.70%	47.90%	47.60%	46%	47.30%
	k=23	k=25	k=27	k=29	
	46.90%	46.90%	45.60%	44.10%	
PCA–KNN	k=3	k=5	k=7	k=9	k=11
	47.60%	50.40%	51.30%	51.90%	50.40%
	k=13	k=15	K=17	k=19	k=21
	50.10%	49.60%	48.90%	48%	49.30%
	k=23	k=25	k=27	k=29	
	48.70%	49.10%	48.90%	49.30%	

### 8. Metode PCA–MKNN dengan PCA–KNN pada 8 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA–MKNN	49%	52.40%	53.60%	52.70%	52.40%
	k=13	k=15	K=17	k=19	k=21
	51.90%	50.90%	50.40%	50%	49.10%
	k=23	k=25	k=27	k=29	
	49.70%	49%	48.10%	47%	
PCA–KNN	k=3	k=5	k=7	k=9	k=11
	48.40%	51.40%	52.60%	54.60%	53.10%
	k=13	k=15	K=17	k=19	k=21
	51.70%	51.90%	53%	52.60%	52.40%
	k=23	k=25	k=27	k=29	
	51.90%	51.70%	53.30%	51.60%	

### 9. Metode PCA–MKNN dengan PCA–KNN pada 9 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA–MKNN	53.90%	54.40%	55.70%	55.70%	54.40%
	k=13	k=15	K=17	k=19	k=21
	52.40%	52.40%	52.70%	52.10%	53.60%
	k=23	k=25	k=27	k=29	
	52.90%	51.70%	51%	49%	

	k=3	k=5	k=7	k=9	k=11
PCA-KNN	54.30%	55.60%	56.10%	56.70%	56.10%
	k=13	k=15	K=17	k=19	k=21
	56%	54.70%	54.90%	55.70%	55.70%
	k=23	k=25	k=27	k=29	
	54.40%	53.70%	54.40%	54.40%	

#### 10. Metode PCA-MKNN dengan PCA-KNN pada 10 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA-MKNN	53.70%	55.30%	55.70%	56.60%	55%
	k=13	k=15	K=17	k=19	k=21
	54.10%	54.30%	53.60%	51.90%	53%
	k=23	k=25	k=27	k=29	
	52.10%	51%	50.30%	48.60%	
PCA-KNN	k=3	k=5	k=7	k=9	k=11
	54.70%	57.30%	56.40%	58.90%	58%
	k=13	k=15	K=17	k=19	k=21
	57.40%	57.30%	56.90%	56.70%	55%
	k=23	k=25	k=27	k=29	
	55.40%	55.10%	53.60%	53.60%	

#### 11. Metode PCA-MKNN dengan PCA-KNN pada 11 fitur baru.

Metode	Nilai k				
	k=3	k=5	k=7	k=9	k=11
PCA-MKNN	56.90%	57.90%	58.30%	59.30%	56.70%
	k=13	k=15	K=17	k=19	k=21
	55.70%	54.60%	55.90%	55.30%	53.90%
	k=23	k=25	k=27	k=29	
	53.10%	52.70%	50.90%	50.70%	
PCA-KNN	k=3	k=5	k=7	k=9	k=11
	57.90%	59.40%	59%	60.30%	59.70%
	k=13	k=15	K=17	k=19	k=21
	59.70%	58.60%	58.30%	58.30%	58.30%
	k=23	k=25	k=27	k=29	
	57.40%	56.40%	55.90%	54%	