**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:

|  |  |  |
| --- | --- | --- |
| Luh Arida Ayu Rahning Putri, S.Kom., M.Cs.  NIP. 198209182008122002 | Ketua Penguji |  |

|  |  |  |
| --- | --- | --- |
| I Wayan Supriana, S.Si., M.Cs.  NIP. 1984082920181113001 | Penguji 1 |  |

|  |  |  |
| --- | --- | --- |
| I Gede Santi Astawa, S.T., M.Cs.  NIP. 198012062006041003 | Penguji 2 |  |

|  |  |  |
| --- | --- | --- |
| I Komang Ari Mogi, S.Kom., M.Kom.  NIP. 198409242008011007 | Penguji 3 |  |

|  |  |  |
| --- | --- | --- |
| Dra. Luh Gede Astuti,M.Kom.  NIP. 196401141994022001 | Penguji 4 |  |

|  |
| --- |
| Mengetahui,  Komisi Seminar dan Tugas Akhir |
| Program Studi Informatika |
| FMIPA UNUD  Ketua, |
|  |
| I Gusti Ngurah Anom Cahyadi Putra, ST., M.Cs |
| NIP. 198403172019031005 |

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

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*. Bertambahnya jumlah musik dalam bentuk digital secara pesat menyebabkan pemberian label *genre* secara manual menjadi tidak efektif. Pemberian label *genre* secara otomatis dapat dilakukan dengan menerapkan algoritma kecerdasan buatan yang dapat mengklasifikasi musik berdasarkan *genre* dengan menggunakan fitur–fitur pada atribut lagu.

Penelitian ini dilakukan untuk mengetahui pengaruh atribut lagu yang digunakan terhadap kinerja untuk mengklasifikasi lagu menggunakan metode algoritma klasifikasi K–Nearest Neighbor (KNN) dan Modified K–Nearest Neighbor (MKNN), dan perbandingan performa yang dihasilkan algoritma KNN dan MKNN pada klasifikasi genre. Penelitian ini menggunakan 12 fitur audio dengan 14 genre pada Spotify, dan juga melakukan seleksi fitur menggunakan Principal Component Analysis (PCA) dengan membandingkan KNN dan MKNN menggunakan *k–fold cross–validation* dan mengukur seberapa tepat klasifikasi terhadap kelas menggunakan *confusion matrix* dengan akurasi pada tiap kelas *genre*.

Dari penelitian ini didapatkan hasil bahwa perbandingan performa yang dihasilkan algoritma KNN dan MKNN tanpa menggunakan seleksi fitur PCA pada klasifikasi 14 genre dengan menggunakan 12 variabel fitur mendapatkan akurasi lebih tinggi, sedangkan jika menggunakan seleksi fitur Principal Component Analysis (PCA), algoritma KNN dan MKNN belum bisa menghasilkan akurasi yang lebih tinggi dari akurasi yang tanpa menggunakan PCA.

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

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

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 streaming media. The rapid increase in the number of music in digital form has made manual genre labeling ineffective. Genre labeling can be done automatically by applying artificial intelligence that can classify music based on genre by using the features in the song attributes.*

*This research was conducted to determine the effect of song attributes used on performance to classify songs using the K – Nearest Neighbor (KNN) and Modified K – Nearest Neighbor (MKNN) classification algorithm methods, and the comparison of the performance produced by the KNN and MKNN algorithms in genre classification. This study uses 12 audio features with 14 genres on Spotify, and also selects features using Principal Component Analysis (PCA) by comparing KNN and MKNN using k-fold cross-validation and measuring how precise the classification of classes is using confusion matrix with accuracy for each. genre class.*

*From this research, it is found that the comparison of the performance produced by the KNN and MKNN algorithms without using PCA feature selection in the 14 genre classification using 12 feature variables gets higher accuracy, whereas when using the Principal Component Analysis (PCA) feature selection, the KNN and MKNN algorithms. has not been able to produce an accuracy higher than that without using PCA.*

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

# **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.

4. 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.

5. 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, 13 Januari 2021

Penyusun

# **DAFTAR ISI**

LEMBAR JUDUL……………………………….………………………………....i

[LEMBAR PENGESAHAN TUGAS AKHIR ii](#_Toc61186420)

[ABSTRAK iii](#_Toc61186421)

[*ABSTRACT* iv](#_Toc61186422)

[KATA PENGANTAR v](#_Toc61186423)

[DAFTAR ISI vi](#_Toc61186424)

[DAFTAR TABEL viii](#_Toc61186425)

[DAFTAR GAMBAR ix](#_Toc61186426)

[BAB I](#_Toc61186427) [PENDAHULUAN 14](#_Toc61186428)

[1.1 Latar Belakang 14](#_Toc61186429)

[1.2 Rumusan Masalah 16](#_Toc61186430)

[1.3 Tujuan Penelitian 16](#_Toc61186431)

[1.4 Batasan Masalah 17](#_Toc61186432)

[1.5 Manfaat Penelitian 17](#_Toc61186433)

[BAB II](#_Toc61186434) [TINJAUAN PUSTAKA 18](#_Toc61186435)

[2.1 Tinjauan Empiris 18](#_Toc61186436)

[2.2 Tinjauan Teoritis 20](#_Toc61186437)

[2.2.1 Lagu dan Musik 20](#_Toc61186438)

[2.2.2 Spotify 26](#_Toc61186439)

[2.2.3 Identifikasi Fitur Audio Spotify 27](#_Toc61186440)

[2.2.4 Transformasi Data 30](#_Toc61186441)

[2.2.5 Klasifikasi 30](#_Toc61186442)

[2.2.6 *Principal Component Analysis* (PCA) 31](#_Toc61186443)

[2.2.7 *K–Fold Cross–Validation* 32](#_Toc61186444)

[2.2.8 *Confusion Matrix* 33](#_Toc61186445)

[2.2.9 *K–Nearest Neighbor* 34](#_Toc61186446)

[2.2.10 *Modified K–Nearest Neighbor* 35](#_Toc61186447)

[BAB III](#_Toc61186448) [METODOLOGI PENELITIAN 37](#_Toc61186449)

[3.1 Pengumpulan Data 37](#_Toc61186450)

[3.2 Variabel dan Definisi Variabel 37](#_Toc61186451)

[3.3 Alur Penelitian 39](#_Toc61186452)

[3.4 Proses KNN (K–*Nearest Neighbor)* 41](#_Toc61186453)

[3.5 Proses MKNN (*Modified K–Nearest Neighbor)* 42](#_Toc61186454)

[3.6 Tahap Pengujian dan evaluasi 44](#_Toc61186455)

[3.7 Evaluasi Sistem 44](#_Toc61186456)

[3.8 Implementasi Sistem 46](#_Toc61186457)

[BAB IV](#_Toc61186458) [HASIL DAN PEMBAHASAN 47](#_Toc61186459)

[4.1 Pengaruh Atribut Lagu yang digunakan terhadap Kinerja untuk mengklasifikasi Lagu menggunakan KNN dan MKNN 47](#_Toc61186460)

[4.2 Transformasi Data 60](#_Toc61186461)

[4.3 Implementasi KNN dan MKNN 63](#_Toc61186462)

[4.3.1 KNN 64](#_Toc61186470)

[4.3.2 MKNN 64](#_Toc61186471)

[4.4 Implementasi Sistem 65](#_Toc61186472)

[4.5 Hasil Pengujian 66](#_Toc61186473)

[4.5.1 Pengujian Pertama 66](#_Toc61186476)

[4.5.2 Pengujian Kedua 98](#_Toc61186483)

[4.6 Analisis Hasil Pengujian 112](#_Toc61186484)

[BAB V](#_Toc61186485) [KESIMPULAN DAN SARAN 115](#_Toc61186486)

[5.1 Kesimpulan 115](#_Toc61186488)

[5.2 Saran 116](#_Toc61186489)

[DAFTAR PUSTAKA 117](#_Toc61186490)

[LAMPIRAN 120](#_Toc61186491)

# **DAFTAR TABEL**

[Tabel 2.1. *Confusion Matrix* 32](#_Toc59538298)

[Tabel 3.1. Variabel Fitur Pengguna 36](#_Toc59538299)

[Tabel 3.2. Variabel Fitur Audio 37](#_Toc59538299)

[Tabel 4.1. Data *Playlist* Musik 52](#_Toc59538298)

[Tabel 4.2. Data Variabel Fitur sebelum ditransformasi 59](#_Toc59538299)

[Tabel 4.3. Data Variabel Fitur setelah ditransformasi 60](#_Toc59538299)

[Tabel 4.4. Data Lagu 14 *Genre* setelah ditransformasi 62](#_Toc59538299)

[Tabel 4.5. Pengacakkan Data 66](#_Toc59538298)

[Tabel 4.6. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3 70](#_Toc59538299)

Tabel 4.7. [Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3 72](#_Toc59538299)

[Tabel 4.8. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8 74](#_Toc59538298)

[Tabel 4.9. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasi ke–8 76](#_Toc59538299)

[Tabel 4.10. Pengaruh Atribut Lagu yang digunakan terhadap kinerja untuk mengklasifikasi Lagu menggunakan KNN dan MKNN 111](#_Toc59538299)

# **DAFTAR GAMBAR**

Gambar 1.1. Keunggulan Spotify (Iriansyah, 2018) 13

Gambar 3.1. Diagram Alur Umum Sistem 38

Gambar 3.2. Diagram Alir Proses KNN 40

Gambar 3.3. Diagram Alir Proses MKNN 41

Gambar 3.4. Lanjutan Diagram Alir Proses MKNN 42

Gambar 3.5. 10*–Fold Cross–Validation* 44

Gambar 4.1. *Playlist* *Genre* *Blues* 46

Gambar 4.2. *Playlist* *Genre* *Classical* 46

Gambar 4.3. *Playlist* *Genre* *Country* 47

Gambar 4.4. *Playlist* *Genre* *EDM* 47

Gambar 4.5. *Playlist* *Genre* *Funk* 47

Gambar 4.6. *Playlist* *Genre* *Gospel* 48

Gambar 4.7. *Playlist* *Genre* *Hip Hop* 48

Gambar 4.8. *Playlist* *Genre* *Jazz* 48

Gambar 4.9. *Playlist* *Genre* *Metal* 49

Gambar 4.10. *Playlist* *Genre* *Pop* 49

Gambar 4.11. *Playlist* *Genre* *R&B* 49

Gambar 4.12. *Playlist* *Genre* *Reggae* 50

Gambar 4.13. *Playlist* *Genre* *Rock* 50

Gambar 4.14. *Playlist* *Genre* *Soul* 50

Gambar 4.15. Plot sebaran *Danceability* 53

Gambar 4.16. Plot sebaran *Valence* 53

Gambar 4.17. Plot sebaran *Energy* 54

Gambar 4.18. Plot sebaran *Tempo* 54

Gambar 4.19. Plot sebaran *Loudness* 55

Gambar 4.20. Plot sebaran *Speechiness* 56

Gambar 4.21. Plot sebaran *Instrumentalness* 56

Gambar 4.22. Plot sebaran *Liveness* 57

Gambar 4.23. Plot sebaran *Accousticness* 57

Gambar 4.24. Plot sebaran *Key* 58

Gambar 4.25. Plot sebaran *Mode* 58

Gambar 4.26. Plot sebaran *Duration* 59

Gambar 4.27. Sebaran *Key* setelah ditransformasi 61

Gambar 4.28. Sebaran *Loudness* setelah ditransformasi 61

Gambar 4.29. Sebaran *Tempo* setelah ditransformasi 61

Gambar 4.30. Sebaran *Duration* setelah ditransformasi 61

Gambar 4.31. Sebelum Normalisasi dengan sebaran data 64

Gambar 4.32. Sesudah Normalisasi dengan sebaran data 65

Gambar 4.33. Akurasi MKNN 67

Gambar 4.34. Perbandingan Akurasi MKNN dengan KNN pada Klasifikasi *Genre* 68

Gambar 4.35. Perbandingan Akurasi 10*–Fold* *Cross*–*Validation* MKNN dengan KNN pada *k* = 5 69

Gambar 4.36. Perbandingan Akurasi 10*–Fold* *Cross*–*Validation* MKNN dengan KNN pada *k* = 7 70

Gambar 4.37. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3 71

Gambar 4.38. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3 73

Gambar 4.39. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8 75

Gambar 4.40. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8 77

Gambar 4.41. Perhitungan *euclidean distance* KNN sejumlah *k* = 5 pada data testing ke-0 78

Gambar 4.42. Kelas Data Hasil Klasifikasi KNN pada data testing ke-0 yang terpilih benar 79

Gambar 4.43. Perhitungan jarak *euclidean distance* KNN sejumlah *k* = 5data testing ke-1 79

Gambar 4.44. Kelas Data Hasil Klasifikasi pada data testing ke-1 yang terpilih salah 80

Gambar 4.45. Perhitungan *euclidean distance* MKNN sejumlah *k* = 5 data testing ke-0 80

Gambar 4.46. Menghitung validasi dan bobot data testing ke-0 tetangga ke-0 81

Gambar 4.47. Menghitung validasi dan bobot data testing ke-0 tetangga ke-1 82

Gambar 4.48. Menghitung validasi dan bobot data testing ke-0 tetangga ke-2 82

Gambar 4.49. Menghitung validasi dan bobot data testing ke-0 tetangga ke-3 82

Gambar 4.50. Menghitung validasi dan bobot data testing ke-0 tetangga ke-4 83

Gambar 4.51. Kelas Data Hasil Klasifikasi MKNN pada data testing ke-0 yang terpilih benar 83

Gambar 4.52. Perhitungan *euclidean distance* MKNN sejumlah *k* = 5 data testing ke-1 83

Gambar 4.53. Menghitung validasi dan bobot data testing ke-1 tetangga ke-0 84

Gambar 4.54. Menghitung validasi dan bobot data testing ke-1 tetangga ke-1 85

Gambar 4.55. Menghitung validasi dan bobot data testing ke-1 tetangga ke-2 85

Gambar 4.56. Menghitung validasi dan bobot data testing ke-1 tetangga ke-3 85

Gambar 4.57. Menghitung validasi dan bobot data testing ke-1 tetangga ke-4 86

Gambar 4.58. Kelas Data Hasil Klasifikasi MKNN pada data testing ke-1 yang terpilih salah 86

Gambar 4.59. Perhitungan *euclidean distance* KNN sejumlah *k* = 7 data testing ke-0 77

Gambar 4.60. Kelas Data Hasil Klasifikasi KNN pada data testing ke-0 yang terpilih benar 72

Gambar 4.61. Perhitungan jarak *euclidean distance* KNN sejumlah *k* = 7 data testing ke-6 73

Gambar 4.62. Kelas Data Hasil Klasifikasi KNN pada data testing ke-6 yang terpilih salah 73

Gambar 4.63. Perhitungan *euclidean distance* MKNN sejumlah *k* = 7 data testing ke-0 74

Gambar 4.64. Menghitung validasi dan bobot data testing ke-0 tetangga ke-0 75

Gambar 4.65. Menghitung validasi dan bobot data testing ke-0 tetangga ke-1 75

Gambar 4.66. Menghitung validasi dan bobot data testing ke-0 tetangga ke-2 75

Gambar 4.67. Menghitung validasi dan bobot data testing ke-0 tetangga ke-3 76

Gambar 4.68. Menghitung validasi dan bobot data testing ke-0 tetangga ke-4 76

Gambar 4.69. Menghitung validasi dan bobot data testing ke-0 tetangga ke-5 76

Gambar 4.70. Menghitung validasi dan bobot data testing ke-0 tetangga ke-6 76

Gambar 4.71. Kelas Data Hasil Klasifikasi MKNN pada data testing ke-0 yang terpilih benar 76

Gambar 4.72. Perhitungan *euclidean distance* MKNN sejumlah *k* = 7 data testing ke-6 77

Gambar 4.73. Menghitung validasi dan bobot data testing ke-6 tetangga ke-0 78

Gambar 4.74. Menghitung validasi dan bobot data testing ke-6 tetangga ke-1 78

Gambar 4.75. Menghitung validasi dan bobot data testing ke-6 tetangga ke-2 79

Gambar 4.76. Menghitung validasi dan bobot data testing ke-6 tetangga ke-3 79

Gambar 4.77. Menghitung validasi dan bobot data testing ke-6 tetangga ke-4 79

Gambar 4.78. Menghitung validasi dan bobot data testing ke-0 tetangga ke-5 76

Gambar 4.79. Menghitung validasi dan bobot data testing ke-0 tetangga ke-6 76

Gambar 4.80. Kelas Data Hasil Klasifikasi MKNN pada data testing ke-6 yang terpilih salah 80

Gambar 4.81. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 1 fitur baru 81

Gambar 4.82. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 2 fitur baru 82

Gambar 4.83. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 3 fitur baru 83

Gambar 4.84. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 4 fitur baru 84

Gambar 4.85. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 5 fitur baru 85

Gambar 4.86. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 6 fitur baru 86

Gambar 4.87. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 7 fitur baru 87

Gambar 4.88. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 8 fitur baru 88

Gambar 4.89. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 9 fitur baru 89

Gambar 4.90. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 10 fitur baru 90

Gambar 4.91. Perbandingan Akurasi PCA*–*MKNN dengan PCA*–*KNN pada 11 fitur baru 91

# **BAB I**

# **PENDAHULUAN**

## **Latar Belakang**

Menurut KBBI (2020), Musik diartikan sebagai: 1) Ilmu dan seni menyusun nada atau suara dalam urutan, kombinasi, dan hubungan temporal untuk menghasilkan komposisi (suara) yang mempunyai kesatuan dan kesinambungan, 2) Nada atau suara yang disusun sedemikian rupa sehingga mengandung irama, lagu, dan keharmonisan (terutama yang menggunakan alat–alat yang dapat menghasilkan bunyi–bunyi itu). Komposisi menjadi penting karena pada alunan musik yang indah tergantung pada penataan atriut musik. Jenis–jenis atribut pada musik ini dapat menghasilkan musik yang berbeda*–*beda dari yang lain.

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.



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 diorganisasi berdasarkan *genre* musik memberikan kemudahan pada pendengar musik untuk mencari musik–musik sejenis yang sesuai dengan referensinya. Bertambahnya jumlah musik dalam bentuk digital secara pesat menyebabkan pemberian label *genre* secara manual menjadi tidak efektif. Pemberian label *genre* secara otomatis dapat dilakukan dengan menerapkan algoritma kecerdasan buatan yang dapat mengklasifikasi musik berdasarkan *genre* dengan menggunakan fitur–fitur pada atribut lagu.

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 *genre* musik tersebut kemudian terbagi lagi menjadi beberapa *playlist*, sehingga total *playlist* yang ada di Spotify berjumlah ribuan. Pembagian musik berdasarkan *genre* ini tentunya untuk mempermudah pengguna Spotify dalam memilih atau menemukan lagu kesukaannya sesuai *genre* maupun *playlist*.

Terdapat beberapa penelitian sebelumnya mengenai pengklasifikasian pada *genre* 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*). Penelitian ini mendapatkan nilai akurasi klasifikasi KNN sebesar 44,8%. Nilai tertinggi ada pada *genre* *Classical* dengan akurasi 100% dan nilai terendah ada pada *genre* *Pop* dengan akurasi 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 dengan 11 fitur audio dengan metode KNN serta membahas metode *k–fold cross–validation*, maka pada penelitian ini penulis ingin menggunakan 12 fitur audio dengan 14 *genre* pada Spotify dengan membandingkan kedua metode algoritma klasifikasi *K–Nearest Neighbor* (KNN) dengan metode algoritma klasifikasi *Modified K–Nearest Neighbor* (MKNN) 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 klasifikassi *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*.

## **Rumusan Masalah**

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

Bagaimana pengaruh atribut lagu yang digunakan terhadap kinerja untuk mengklasifikasi Lagu menggunakan KNN dan MKNN?

1. Bagaimana perbandingan performa yang dihasilkan algoritma KNN dan MKNN pada klasifikasi *genre*?

## **Tujuan Penelitian**

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

1. Untuk mengetahui pengaruh atribut lagu yang digunakan terhadap kinerja untuk mengklasifikasi lagu menggunakan KNN dan MKNN.

Untuk mengetahui perbandingan performa yang dihasilkan algoritma KNN dan MKNN pada klasifikasi *genre*.

## **Batasan Masalah**

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

1. 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*.
2. Variabel fitur pada atribut lagu yang digunakan adalah 12 fitur, antara lain *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode,* dan *duration.*
3. Perbandingan metode yang digunakan adalah metode algoritma K–*Nearest Neighbor* dan *Modified K–Nearest Neighbor*.

## **Manfaat Penelitian**

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

1. Bagi Masyarakat

Penelitian ini dapat membantu untuk mendapatkan informasi tentang pengaruh atribut lagu yang digunakan terhadap akurasi untuk mengklasifikasi lagu menggunakan KNN dan MKNN.

1. Bagi Penulis

Penelitian ini diharapkan dapat menambah pengetahuan bagi penulis mengenai perbandingan performa algoritma algoritma KNN dan MKNN pada klasifikasi musik diajukan.

1. Bagi Keilmuan

Penelitian ini dapat menjadi referensi perbandingan performa 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).

## **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.

1. Penelitian tugas akhir oleh Supriyadi (2018) menggunakan lagu dengan *genre* musik *pop* dan klasik pada layanan *streaming music* Spotify dengan mengimplementasikan *Artificial Neural Network* (ANN)

Dalam penelitiannya, metode *Backpropagation* guna untuk melakukan pengklasifikasian *genre* musik pop dan klasik. Data input 7 fitur audio yang digunakan adalah *accousticness*, *danceability*, *energy*, *loudness*, *speechiness*, *tempo* dan *valence* kemudian outputnya adalah *genre* musik *pop* dan klasik. Dari hasil implementasi 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 pada layanan *streaming* Spotify.

Perbedaannya, jika pada penelitian Supriyadi (2018) menggunakan algoritma *Backpropagation*, maka pada penelitian ini penulis menggunakan algoritma KNN dan MKNN.

1. Pada jurnal 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, Gingivtis, 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 mengunakan metode klasifikasi *Modified K–Nearest Neighbor* (MKNN).

Perbedaannya, jika pada penelitian Supriyadi (2018) melakukan klasifikasi identifikasi jenis penyakit gigi dan mulut, maka pada penelitian ini penulis melakukan klasifikasi musik pada layanan *streaming* Spotify.

1. Pada jurnal Parvin, dkk (2010), dilakukan pengujian untuk membandingkan *K–Nearest Neighbor* (KNN) dengan *Modified K–Nearest Neighbor* (MKNN) dengan empat dataset yaitu (a) Iris (b) Balance scale (c) Bupa (d) SAHeart dan nilai *k* yang berbeda. Hasil dari penelitian adalah akurasi *k* = 3, 5 dan 7. Metode *Modified K–Nearest Neighbor* (MKNN) lebih baik dari *K–Nearest Neighbor* (KNN).

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

1. Dananjaya dkk (2019) 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 dilakukan 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%.

## **Tinjauan Teoritis**

### **Lagu dan Musik**

Menurut KBBI (2020), Musik diartikan sebagai: 1) Ilmu dan seni menyusun nada atau suara dalam urutan, kombinasi, dan hubungan temporal untuk menghasilkan komposisi (suara) yang mempunyai kesatuan dan kesinambungan, 2) Nada atau suara yang disusun sedemikian rupa sehingga mengandung irama, lagu, dan keharmonisan (terutama yang menggunakan alat–alat yang dapat menghasilkan bunyi–bunyi itu).

Lagu merupakan ragam suara yang berirama (dalam bercakap, bernyanyi, membaca, dan sebagainya). Selain itu lagu juga sering dikenal dengan suatu syair atau lirik yang mempunyai irama, sebuah lagu biasanya selalu diiringi dengan alat musik untuk menghasilkan musik yang mengandung suara yang berirama. Secara umum lagu memang tidak lepas dengan musik. Pengertian seni musik adalah suatu yang membuahkan hasil karya seni, berupa bunyi berbentuk lagu atau komposisi yang mengungkapkan pikiran serta perasaan penciptanya lewat unsur–unsur pokok musik, yakni irama, melodi, harmoni, tempo, birama, timbre, tangga nada, dinamika serta ekspresi sebagai satu kesatuan susunan lagu (Jamalus, 1988).

Dari sekian 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 bunyi musik, berapa lama musik tersebut, dan bagaimana pendengar harus berperilaku. Pada era *Mozart*, ada lima *genre* utama, yaitu *symphony, string quartet, sonata, concerto,* dan opera. 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* berkembang dari musik–musik spiritual dan puji–pujian yang muncul dari komunitas budak–budak Afrika di AS silam. Sebelum abad ke–20, musik *blues* hanya populer di kalangan orang Amerika. Ciri – ciri musik ini yaitu pola irama yang sering digunakan, terkesan sedih dan permainan gitarnya selalu penuh dengan improvisasi.

1. Musik *Classical*

Musik *classical*atau klasik adalah kumpulan dari *genre* musik pada era Mozart dengan bentuk seperti *symphony, string quartet, sonata, concerto,* dan opera. Musik *classical* biasa dimainkan dengan alat–alat musik tertentu seperti biola, beranjak pada abad ke–9 hingga abad ke–21 dan biasanya merujuk pada musik klasik Eropa. Ciri–ciri musik klasik yakni penggunaan dinamika *crescendo* dan *decrescendo*. *Crescendo* yaitu perubahan dinamika lagu dari lembut menjadi keras dan nyaring. Sedangkan *decrescendo* sebaliknya, perubahan dinamikanya dari keras menjadi lembut. Ciri lainnya yaitu perubahan *tempo* nya menggunakan *accelerando* (semakin cepat) dan *ritardando* (semakin lembut).

1. Musik *Country*

Musik *Country* adalah campuran dari unsur–unsur musik Amerika yang berasal dari 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 dipakai sekitar tahun 1940–an untuk menggantikan istilah musik *hillbilly* yang berkesan merendahkan. Pada tahun 1970–an, istilah musik *country* telah menjadi istilah populer. Istilah lain untuk *genre* musik ini adalah *country* and *western*, namun sudah semakin jarang dipakai kecuali di Britania Raya dan Irlandia. Contoh orang yang menggunakan *Genre* *Country*: Taylor Swift.

1. Musik *Gospel*

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

Ciri khas dari *genre* *Gospel* adalah vokal yang dominan dengan harmoni yang kuat dan liriknya yang religius, khususnya Kristen. *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 *Gospel* yang sudah dikombinasikan dengan *genre* musik lain. Kemudian diikuti oleh Golden Gate Quartet dan Clara War yang lebih berani membawakan tembang–tembang gereja ini di club malam dalam performnya. Hingga kini Musik *Gospel* sudah lagi tidak ada perbedaan yang kentara dengan musik kontemporer.

1. Musik *Hip Hop*

Musik *Hip Hop* merupakan salah satu *genre* musik yang terdiri dari perpaduan *rapping, DJing, Breakdance* dan *Graffiti*. *Hip Hop* mulai ada sekitar tahun 1970–an yang dipelopori oleh kalangan Afro–Amerika dan Amerika Latin. Awalnya *Hip Hop* hanyalah musik dari *Disk Jockey* yang membuat variasi dengan memutar disk sehingga menciptakan bunyi yang unik. Kemudian *Rapping* masuk untuk menghiasi vokal dari bunyi–bunyi tersebut. Sedangkan untuk 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.

1. Musik *Jazz*

Musik *jazz* merupakan jenis musik yang tumbuh dari penggabungan *blues, ragtime,* dan musik eropa, terutama musik band. Aliran musik ini berasal dari masyarakat Afro–Amerika Selatan pada akhir abad ke–19 dan awal abad ke–20. Kata *Jazz* berasal dari bahasa slang (bahasa daerah pinggiran pantai barat Amerika Serikat) dan pertama kali dipakai istilah *jazz* pada tahun 1915 di Chicago. Ciri–ciri dari *genre* ini yaitu banyak menggunakan instrumen gitar, trombone, terompet, saksofon dan vokal liriknya cenderung dianggap sebagai bagian dari bunyi instrumen, ritme dan melodinya memiliki cenderung mengimprovisasi nada.

1. Musik *Metal*

Musik *Metal* mengutamakan permainan gitar dan biasanya dimainkan dengan *tempo* lagu yang sangat cepat. Musik *Metal* berawal dari kata Heavy *Metal* yang terdapat dari lagu band Hard *Rock* 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 bercirikan solo gitar yang panjang, distorsi gitar yang kuat dan ketukan yang cepat di semua instrumen alat musiknya.

*Heavy metal* pada era ini melahirkan band–band besar seperti Led Zeppelin, Black Sabbath dan Deep Purple. Namun elemen *blues* masih terlihat kental mempengaruhi di era ini. Selanjutnya Judas Priest mengimprovisasi *genre* ini dengan meniadakan unsur *blues* nya, sehingga hanya mengandalkan distorsi, beat yang lebih cepat dan harmoni.

Perkembangan demi perkembangan terus terjadi dalam sejarah salah satu jenis musik ini sampai akhirnya menjadikannya terbagi menjadi beberapa sub–*genre*, diantaranya *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*.

1. Musik *Pop*

Musik *pop* atau musik populer merupakan jenis–jenis musik yang saat ini digemari oleh masyarakat awam. *Genre* ini mulai berkembang dan populer pada tahun 1950–an. *Genre* dinamakan populer karena memiliki daya tarik masa yang lebih besar dibandingkan dengan *genre* lainnya. Ciri– ciri musik *pop* yaitu 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.

1. Musik *Reggae*

Musik *Reggae* mempunyai ciri khas dengan ritme backbeat dan progresif kord sederhana. *Genre* musik ini berasal dari Jamaika dan mirip dengan gaya musik ska dan *rock*steady. 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. Dahulu musik reggae sering dianggap dengan jiwa muda, *rude boys* atau anak laki–laki kasar dan pemberontakan. Dan kita semua tahu, *genre* musik ini melahirkan beberapa musisi hebat seperti Bob Marley, Prince Buster, Desmond Dekker dan Jackie Mittoo sebagai legendanya meskipun sebetulnya masih banyak lagi yang lain.

1. Musik *Rock*

Lahirnya musik *Rock* berawal dari kombinasi dari *genre* musik *rhythm* dan *blues*, *country*, *jazz*, musik klasik termasuk musik rakyat (*folk music*). Jenis musik ini mulai dikenal secara umum pada pertengahan tahun 1950–an. *Rock* 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*](https://www.onoini.com/pengertian-musik-jazz/)*rock, New wave, Pop rock, Post britpop, Post grunge, Post punk, Progressive rock, Psychedelic rock, Punk rock, Rock and roll, Rockabily, 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.

1. 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. 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.

1. Musik *Funk*

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

1. Musik *Rhythm* and *Blues* (R&B)

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

Periode awal musik *R&B* berfokus pada ritme boogie dengan musisi ternama seperti Bo Diddley dan Chuck Berry. Sementara 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 fokus pada unsur *pop* (walaupun elemen *jazz*, *gospel* dan *blues* terkadang masih tetap muncul). Dari waktu ke waktu, kesan *genre* *R&B* sebagai musik kalangan kulit hitam semakin memudar dengan menyebarnya musik ini ke seluruh dunia.

1. Musik *Soul*

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

### **Spotify**

Spotify merupakan layanan *streaming music* asal Swedia yang menyediakan 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. Selain itu, Spotify juga berperan sebagai penyedia hak cipta digital atas musik yang diunggah, sehingga lagu–lagu yang disajikan 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* (Spotify, 2020).

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.

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.

1. *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.

1. Radio

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

1. 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.

1. 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.

### **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.

1. *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.

1. *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.

1. *Tempo*

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

1. *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.

1. *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*.

1. *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.

1. *Liveness*

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

1. *Acousticness*

Suatu ukuran yang menunjukan 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 menunjukan semakin tinggi kepercayaan lintasannya akustik, begitu pula sebaliknya apabila semakin mendekati 0.0 maka semakin rendah kepercayaan lintasnya akustik.

1. *Key*

Taksiran keseluruhan kunci trek. *Integer* memetakan ke pitch menggunakan notasi *Pitch Class*. Misalnya 0 = C, 1 = C♯**/**D♭, 2 = D, dan seterusnya.

1. *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.

1. *Duration*

Durasi trek musik dalam milidetik.

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

### **Transformasi Data**

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

...………... (2.1)

Keterangan:

*V’* = Nilai yang dicari.

*V* = Nilai pada variabel A.

𝑀𝑖𝑛𝐴 = Nilai minimal variabel A.

𝑀𝑎𝑥𝐴 = Nilai maksimal variabel A.

𝑁𝑒𝑤\_𝑚𝑖𝑛𝐴 = Rentang nilai minimal pada variabel A.

𝑁𝑒𝑤\_𝑚𝑎𝑥𝐴 = Rentang nilai maksimal pada variabel A.

### **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*.

### ***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).

................. (2.2)

1. Menghitung *covariance*

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

................. (2.3)

1. Menghitung *eigenvalue* dan *eigenvectors*

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

................. (2.4)

................. (2.5)

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

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

................. (2.6)

Keterangan:

: *Matrix* n x n.

: *Eigenvalue*.

: *Identity Matrix*.

: Korelasi antara variabel utama dengan komponen utama.

: *eigenvectors*.

: *Covariance Matrix*.

### ***K–Fold Cross–Validation***

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

### ***Confusion Matrix***

*Confusion Matrix* adalah tabel yang sering digunakan untuk menggambarkan kinerja model pada klasifikasi (atau "*classifier*") pada serangkaian data *testing* (data uji) yang nilai sebenarnya diketahui (Nomleni, 2015). Tabel 2.1 menunjukkan *confusion matrix* untuk *classifier*.

Tabel 2.1. Confusion Matrix

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

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

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

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

..………………............................ (2.7)

### ***K–Nearest Neighbor***

Metode Algoritma *K–Nearest Neighbor* (KNN) yaitu algoritma yang bertujuan untuk klasifikasi data. Secara umum, metode ini digunakan untuk menentukan kelas data dari suatu data dengan mengukur jarak antar tetangganya (Widiantara, 2018). Jarak yang dipilih adalah jarak terpendek dari data *testing* (data uji). Perbandingan jarak antara data *testing* (data uji) dan data *training* (data latih) dilakukan sebanyak *k* buah. Rumus perhitungan jarak tersebut dapat dilakukan dengan *euclidean distance* dituliskan pada Persamaan (2.8).

..................................... (2.8)

Keterangan:

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

: Atribut fitur data latih.

: Atribut fitur data uji.

*i* : Mempresentasikan index nilai atribut fitur.

*n*  : Jumlah panjang dimensi atribut fitur.

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

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

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

### ***Modified K–Nearest Neighbor***

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

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

…………..……...……………..... (2.9)

Keterangan:

: Nilai validitas.

: Besaran atau jumlah titik terdekat dari individu.

: Kelas data latih.

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

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

…..……………………................................. (2.10)

Keterangan:

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

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

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

…………..……............................ (2.11)

Keterangan:

: Bobot data latih ke–i.

: Validitas data latihke–i.

: Jarakantara data latihke–i dengan data uji.

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

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

…………………….……............................. (2.12)

Keterangan:

: *Voting* dari kelas data latih x.

: Jumlah data latihdengan kelas data x.

: Nilai indeks.

: Bobot data latihke–i.

# **BAB III**

# **METODOLOGI PENELITIAN**

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

## **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.

## **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, accousticness, key, mode,* dan *duration*). Penjelasan dan definisi dari variabel yang digunakan dalam penelitian ini ditampilkan dalam Tabel 3.1 dan Tabel 3.2.

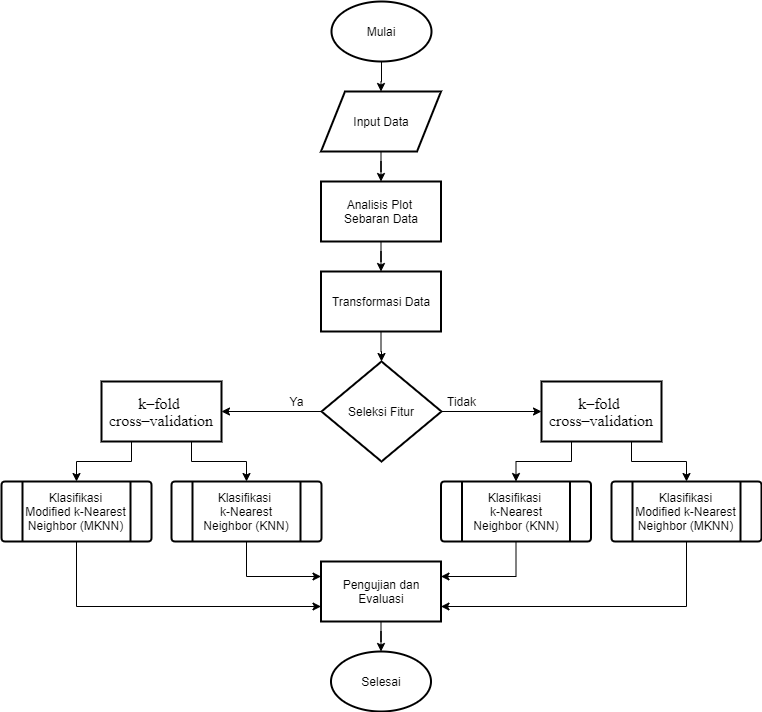
Tabel 3.1. Variabel Fitur Pengguna

|  |  |  |  |
| --- | --- | --- | --- |
| No | Variabel | Definisi Variabel | Ukuran |
| 1. | *Playlist* | *Playlist* adalah atribut yang didalamnya berisikan data lagu. *Playlist* akan dibuat untuk setiap *genre*. Misalkan daftar *playlist* 0 untuk *genre* musik *Blues*, 1 untuk *genre* musik *Classical*, 2 untuk *genre* musik *Country,* dan seterusnya | – |

Tabel 3.2. Variabel Fitur Audio

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | *Danceability* | Menjelaskan seberapa kecocokan sebuah lagu atau trek musik untuk menari | 0,0 – 1,0 |
| 2. | *Valence* | Mengindikasi jenis emosi (seperti bahagia atau sedih) | 0,0 – 1,0 |
| 3. | *Energy* | Ukuran intensitas dan aktivitas suatu tingkat dari kemampuan suatu musik untuk meningkatkan emosi dari pendengarnya | 0,0 – 1,0 |
| 4. | *Tempo* | Tempo merupakan cepat atau lambatnya sebuah lagu. Ukuran dari sebuah *tempo* adalah *beats per minute* (BPM). | *Beats Per Minute* (BPM) |
| 5. | *Loudness* | Kenyaringan (lembut) pada suatu lagu dalam desibel (dB) | –60 – 0 Db |
| 6. | *Speechiness* | 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. | *Instrumentalness* | Memprediksi apakah suatu lagu mengandung vokal atau tidak, seperti suara “Ooh” dan “Aah” | 0,0 – 1,0 |
| 8. | *Liveness* | Mendeteksi keberadaan *audiens* dalam rekaman | 0,0 – 1,0 |
| 9. | *Acousticness* | Suatu ukuran yang menunjukan apakah trek tersebut akustik atau tidak | 0,0 – 1,0 |
| 10. | *Key* | Taksiran keseluruhan kunci trek. *Integer* memetakan ke *pitch* menggunakan notasi *Pitch Class* standar | 0 = C,  1 = C♯**/**D ♭,  2 = D |
| 11. | *Mode* | Mengindikasikan modalitas (besar atau kecil) dari suatu trek, jenis skala dari mana konten melodinya diturunkan | Mayor diwakili oleh 1 dan minor adalah 0 |
| 12. | *Duration* | Durasi trek music | Milidetik |

## **Alur Penelitian**



Gambar 3.1. Diagram Alur Umum Sistem

Pada bagian ini akan digambarkan alur secara umum dari penelitian yang akan dilakukan penulis seperti pada Gambar 3.1, yaitu dimulai dari pengumpulan data dari web api Spotify, lalu dilakukan analisis plot sebaran data, transformasi data, pengujian, pembagian data *training* (data latih) dan data *testing* (data uji)dengan k*–fold cross–validation*, lalu klasifikasi dengan metode KNN dan MKNN, tahap pengujian dan evaluasi sistem. Berikut ini definisi penjelasan dari masing–masing tahap alur penelitian pada Gambar 3.1.

Tahapan pertama yang dilakukan dalam penelitian ini adalah 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*.

Tahapan kedua yang dilakukan dalam penelitian ini yakni analisis *plot* sebaran data atribut lagu berdasarkan variabel pada Tabel 3.1 dan Tabel 3.2 pada *genre* musik yang digunakan.

Tahap ketiga 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 menggunakan cara pada Persamaan (2.1).

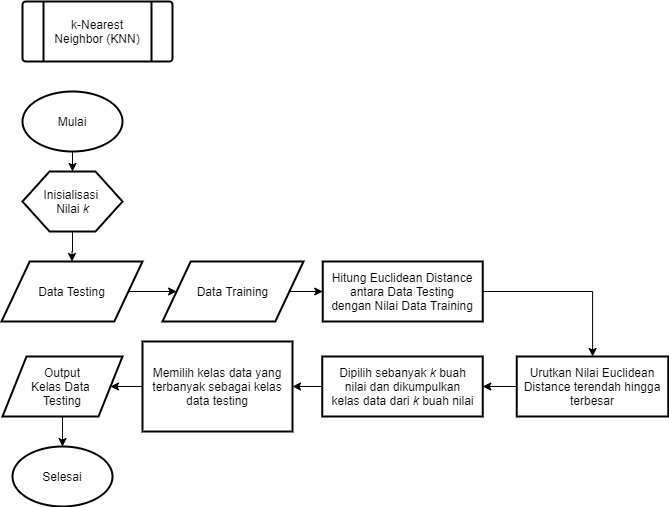
Tahap keempat adalah pengujian. Pengujian akan dilakukan dengan dua pengujian. Pada pengujian pertama yaitu tanpa seleksi fitur *Principal Component Analysis* (PCA) dan pengujian kedua yaitu menggunakan seleksi fitur *Principal Component Analysis* (PCA). Untuk pengujian pertama akan langsung ke tahap kelima yaitu menggunakan metode *k–fold cross–validation*. Sedangkan pengujian kedua akan sama menggunakan menggunakan metode *k–fold cross–validation* setelah melakukan seleksi fitur.

Tahap kelima adalah 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.

Tahapan keenam yang dilakukan dalam penelitian ini adalah proses pengklasifikasian yang dilakukan menggunakan metode *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN).

Tahap terakhir adalah pengujian dan evaluasi sistem yang telah dibuat. Dalam penelitian ini akan dilakukan pengujian terhadap sistem yang telah dibangun dengan mengklasifikasi *genre* menggunakan algoritma KNN dan MKNN dengan seleksi fitur *Principal Component Analysis* (PCA) dan tanpa seleksi fitur *Principal Component Analysis* (PCA) menggunakan *k–fold cross–validation* dalam menentukan *k* terbaik dan *confusion matrix*.

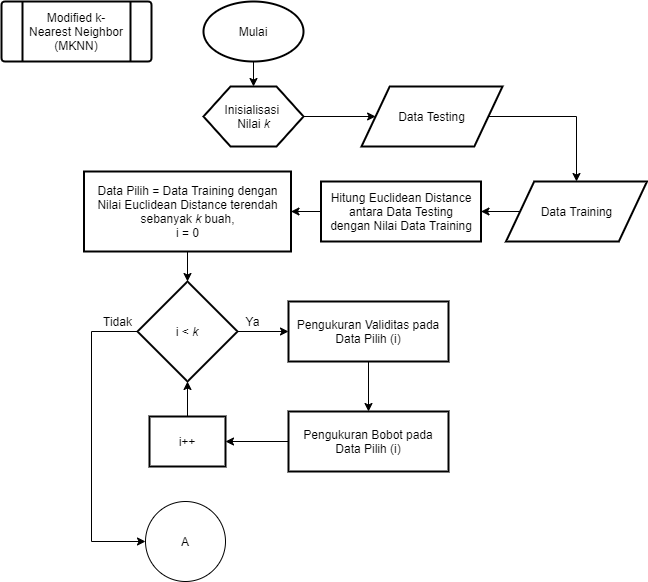
## **Proses KNN (K–*Nearest Neighbor)***



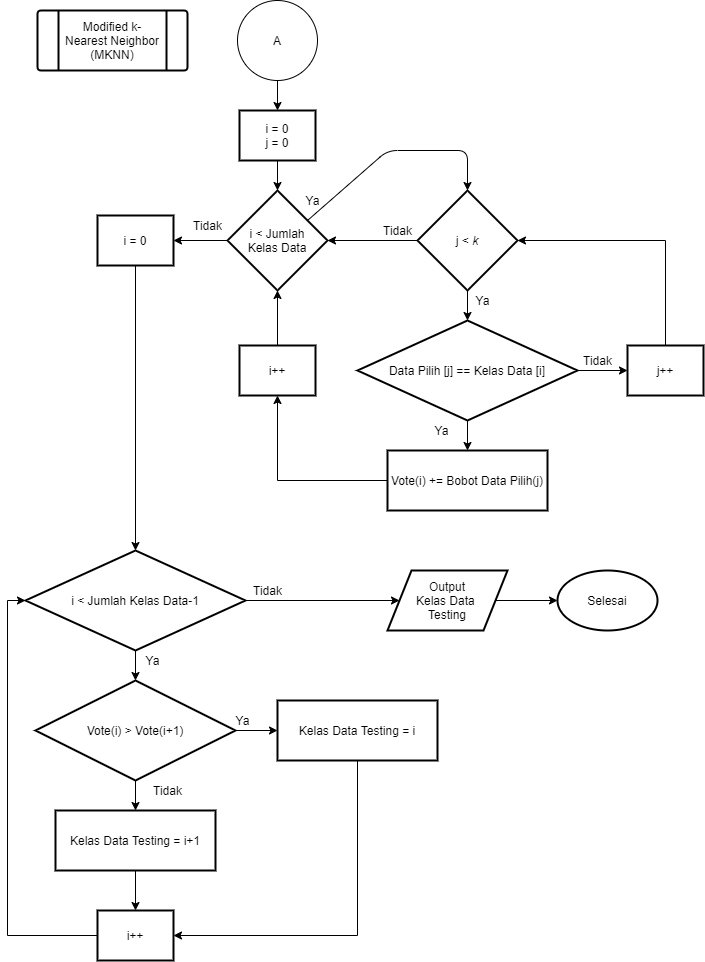
Gambar 3.2. Diagram Alir Proses KNN

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.

## **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.

## **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**

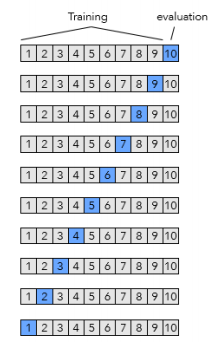
Pada pengujian pertama dalam penelitian ini yaitu membandingkan akurasi dari tiap perubahan nilai *k* pada *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan metode *k–fold cross–validation* tanpa seleksi fitur *Principal Component Analysis* (PCA).

**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).

## **Evaluasi Sistem**

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, dipilih sebagai data uji, dan partisi yang tersisa yaitu ,….,secara kolektif digunakan sebagai data latih. Sedangkan dalam iterasi kedua, partisi dipilih sebagai data uji, dan partisi , ,…,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 data uji akan dibandingkan dengan data latih untuk menentukan akurasi, sehingga akurasi total dapat diukur berdasarkan jumlah keseluruhan akurasi klasifikasi yang benar dari iterasi *k–fold* dibagi dengan jumlah total *k–fold* yang digunakan. Rumus yang digunakan sebagai berikut (lihat pada Persamaan 2.14).

............. (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.

## **Implementasi Sistem**

Sistem diimplementasikan menggunakan bahasa pemrogramman *Pyhton* dengan versi 3.8.2 serta menggunakan IDE Visual Studio Code. Penelitian ini juga menggunakan software Microsoft Excel Plus 2019 dalam menyimpan dataset. Sistem ini dijalankan dengan menggunakan Sistem Operasi Windows 10 64-bit.

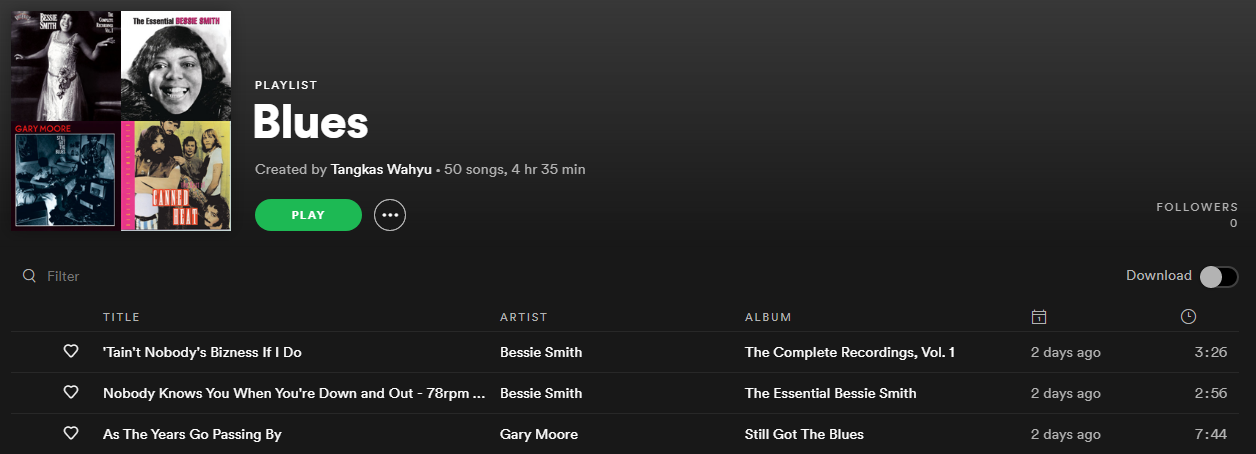
# **BAB IV**

# **HASIL DAN PEMBAHASAN**

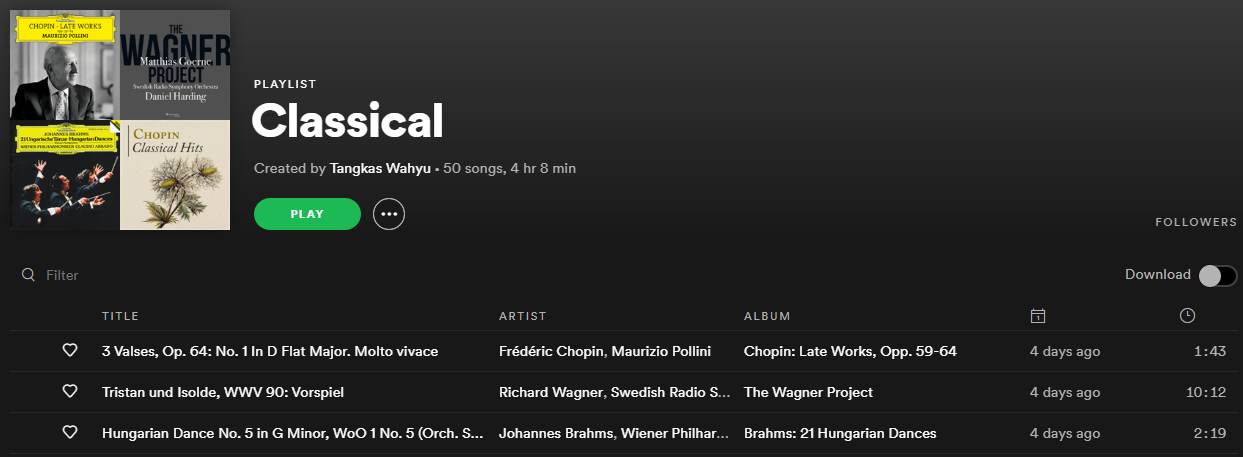
Pada bab ini, akan dijelaskan mengenai hasil pengaruh atribut lagu terhadap kinerja KNN dan MKNN, transformasi data, implementasi KNN dan MKNN, implementasi sistem, pengujian pertama klasifikasi *genre* tanpa seleksi fitur, pengujian kedua dengan seleksi fitur dan analisis hasil pengujian.

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

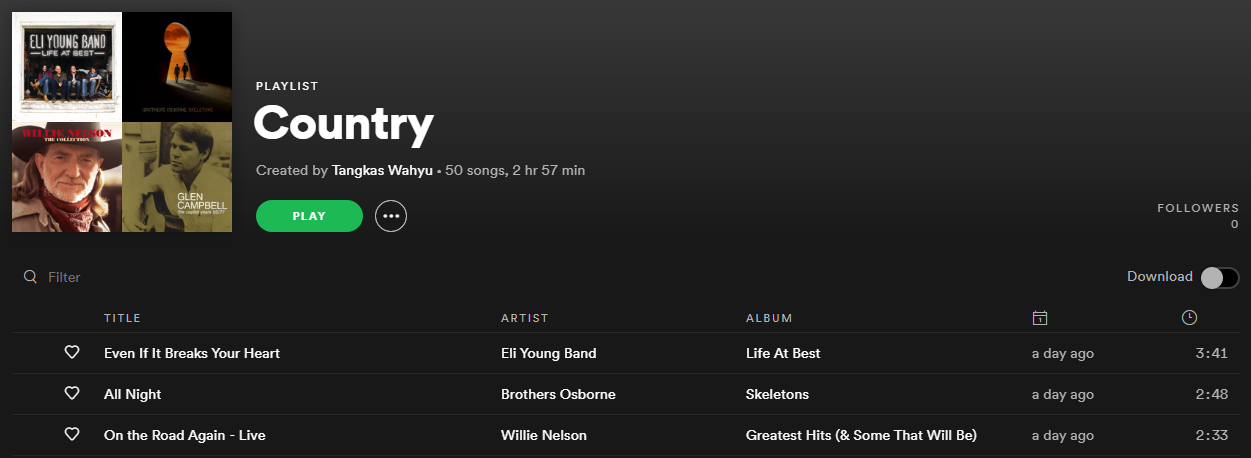
Spotify digunakan sebagai platform dalam membuat *playlist* *genre* berisi lagu. Pada penelitian ini, *playlist* baru dibuat agar dapat dibedakan dengan *playlist* yang telah ada. Sebuah lagu kemudian diambil pada *playlist* yang sudah disediakan oleh Spotify. Berikut adalah *playlist* 14 *genre* musik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



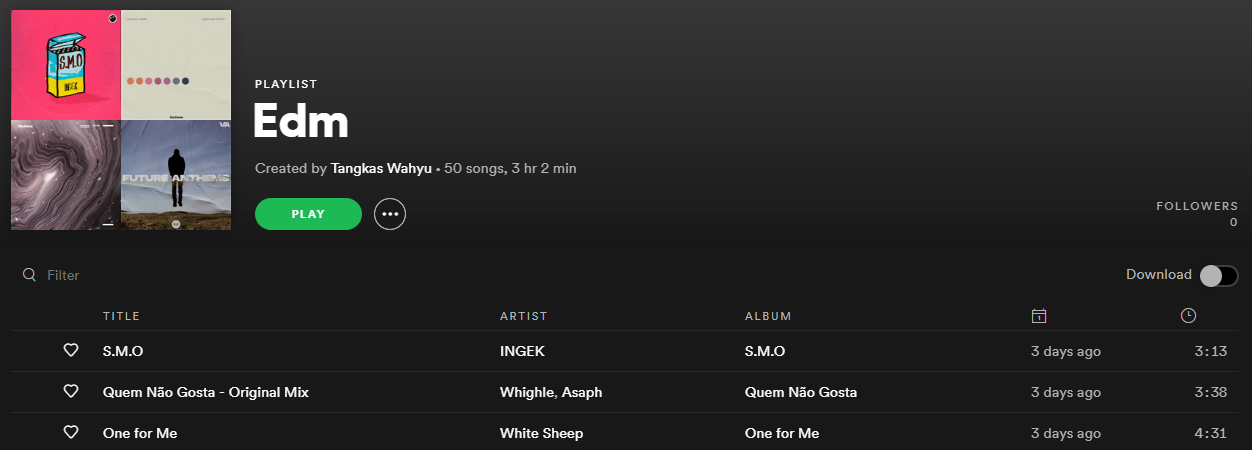
Gambar 4.1. Playlist Genre Blues



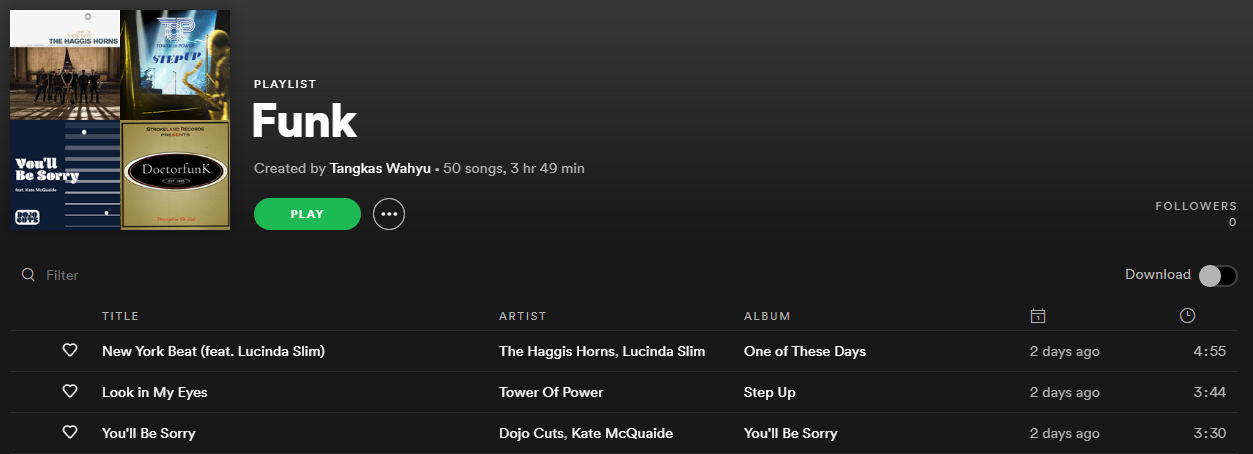
Gambar 4.2. *Playlist Genre Classical*



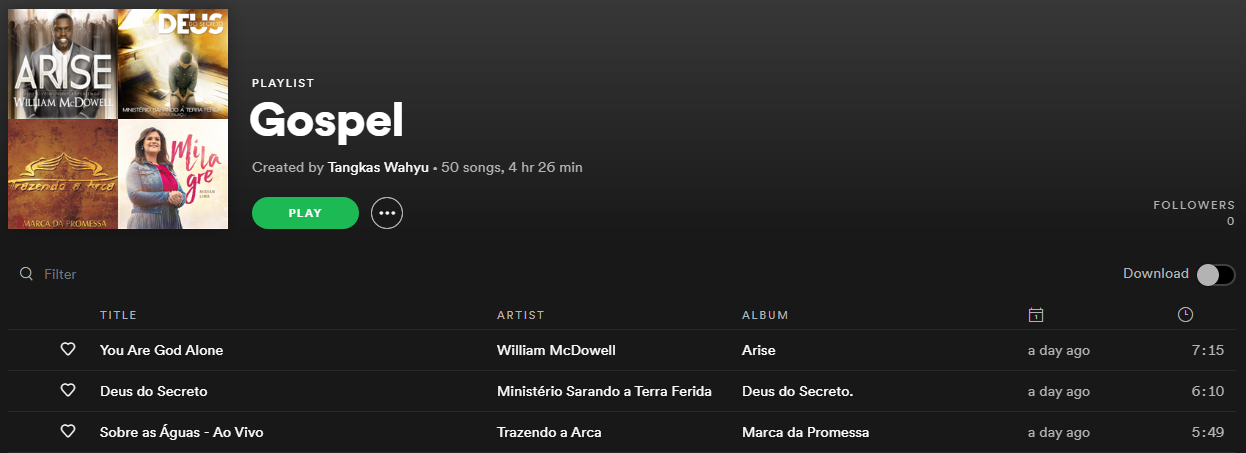
Gambar 4.3. Playlist Genre Country



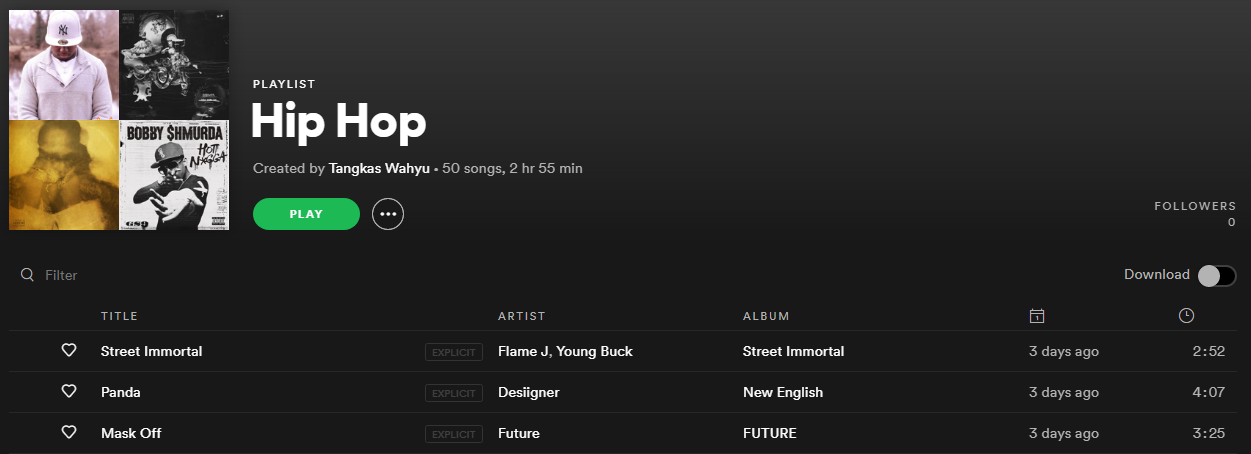
Gambar 4.4. Playlist Genre EDM



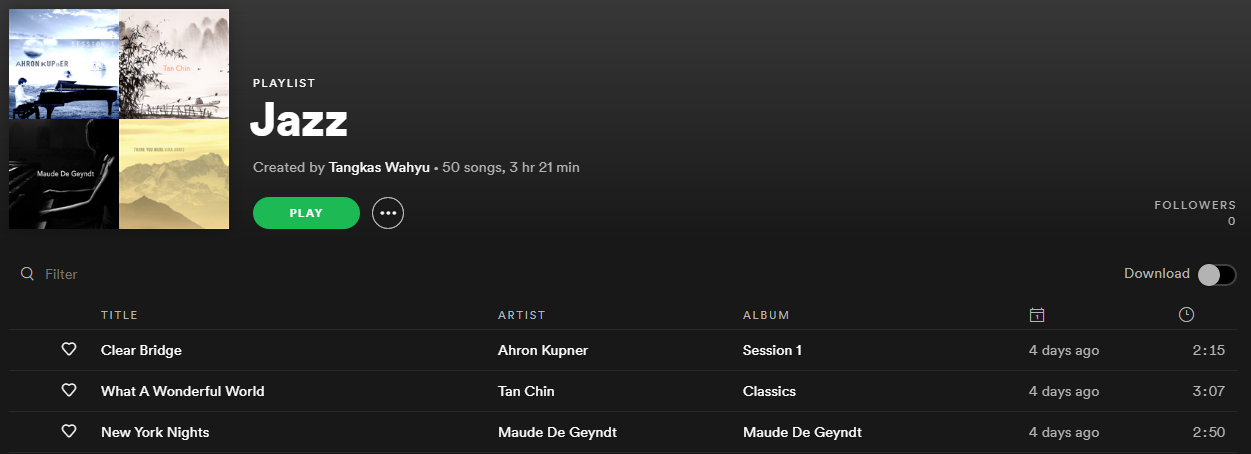
Gambar 4.5. Playlist Genre Funk



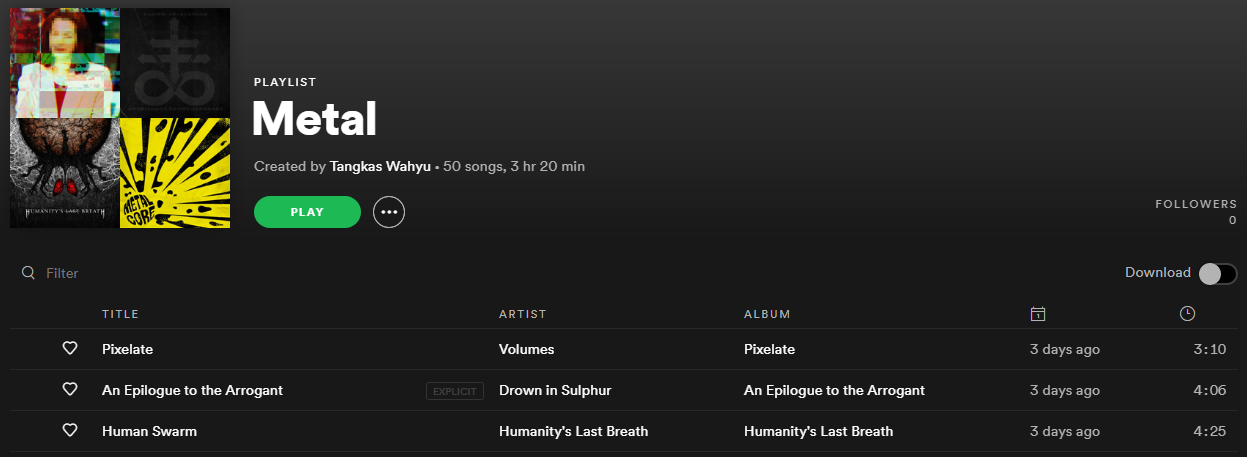
Gambar 4.6. Playlist Genre Gospel



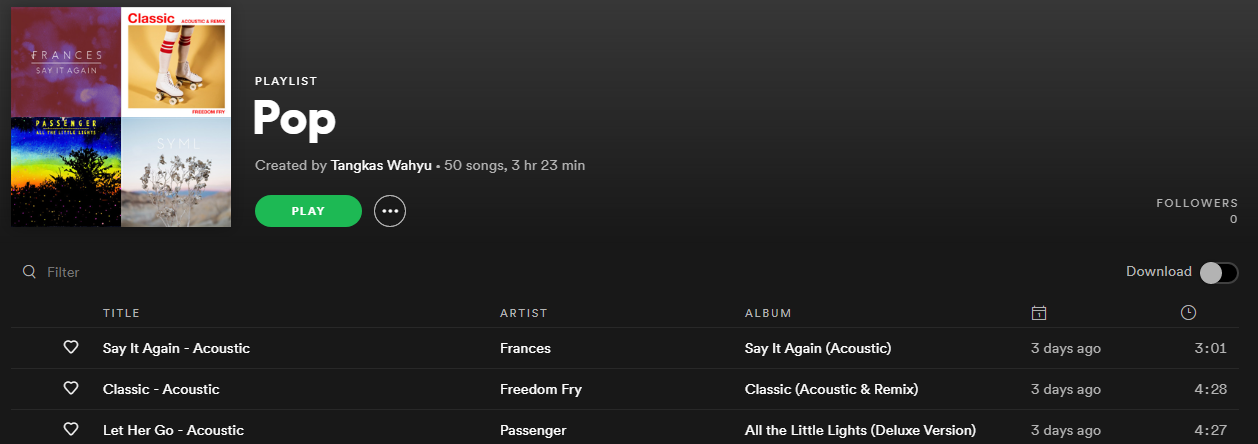
Gambar 4.7. Playlist Genre Hip Hop



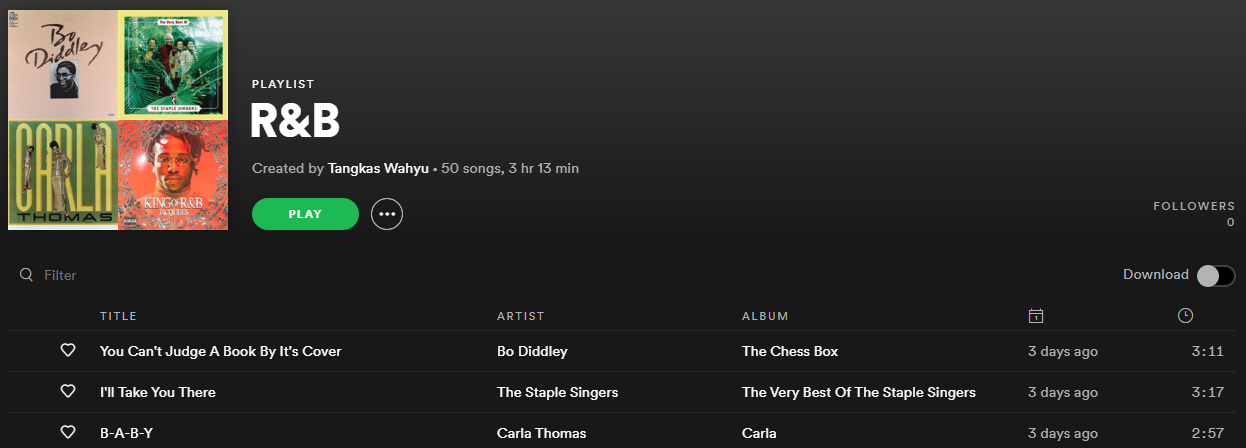
Gambar 4.8. Playlist Genre Jazz



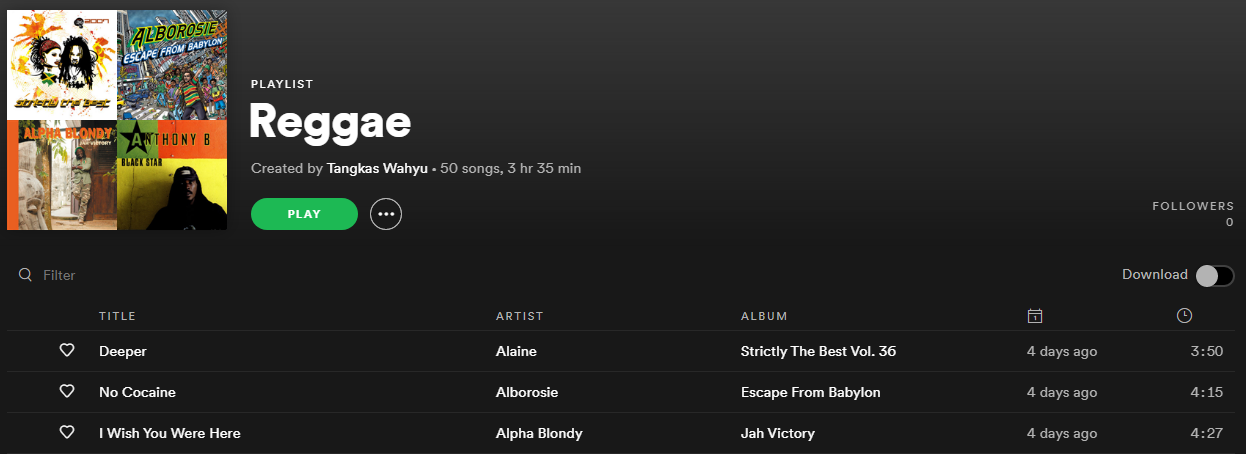
Gambar 4.9. Playlist Genre Metal



Gambar 4.10. Playlist Genre Pop



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 3.8. 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'] = '9a8cb565889548e69d001505e9940d4b'

os.environ['SPOTIPY\_CLIENT\_SECRET'] = 'c251cd9f1eaf427081525bf4bd774bea'

sp = spotipy.Spotify(client\_credentials\_manager=SpotifyClientCredentials()) # 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 = sp.audio\_features(Panjang track = track\_list)

pd.DataFrame(features).to\_csv('**Lokasi Penyimpanan dan nama file.csv**') # Save to spreadsheet format .csv

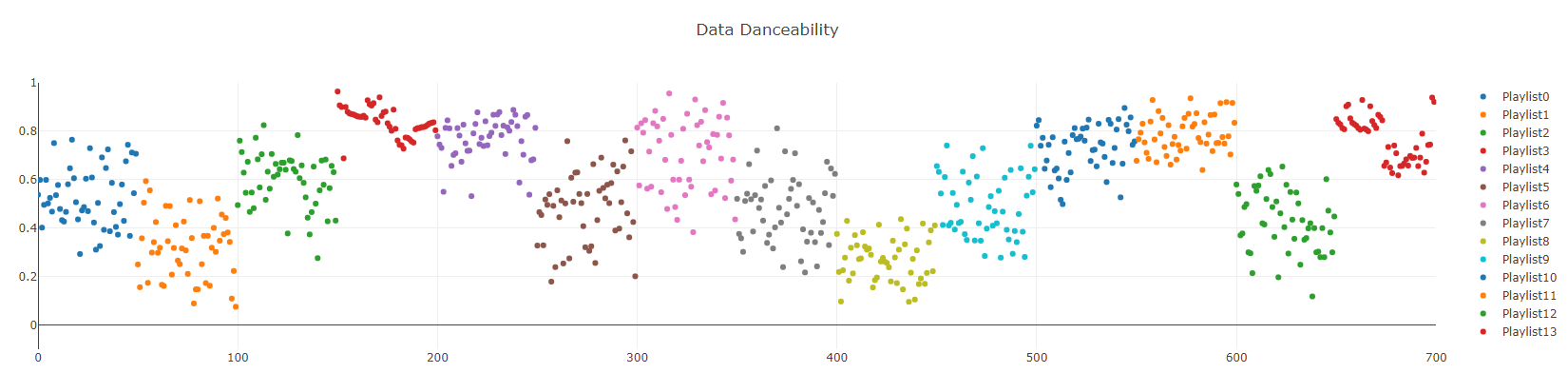
14 *playlist* yang dibuat, akan di beri label pada *playlist* tersebut, agar bisa membedakan antar *genre*. Untuk *genre* *Blues* akan diberi label 0, *genre* *Classical* diberi label 1, *genre* *Country* diberi label 2, *genre* *EDM* diberi label 3, *genre* *Funk* diberi label 4, *genre* *Gospel* diberi label 5, *genre* *Hip Hop* diberi label 6, *genre* *Jazz* diberi label 7, *genre* *Metal* diberi label 8, *genre* *Pop* diberi label 9, *genre* *R&B* diberi label 10, *genre* *Reggae* diberi label 11, *genre* *Rock* diberi label 12, dan *genre* *Soul* diberi label 13.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 ditunjukan 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 |

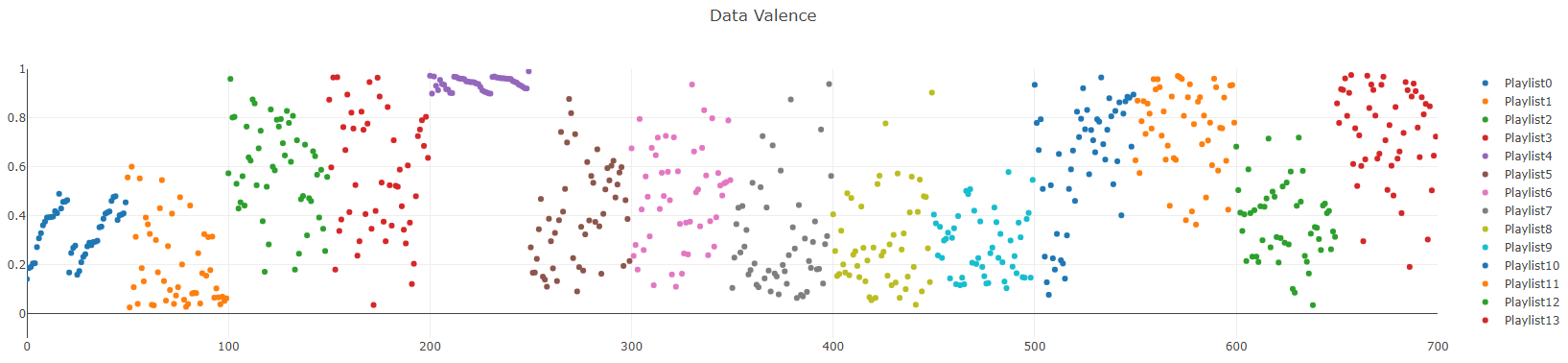
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Playlist | Instrumentalness | Liveness | Valence | Tempo | Duration\_ms |
| 1 | 0 | 0.000606 | 0.212 | 0.141 | 80.659 | 206227 |
| 2 | 0 | 0.00176 | 0.185 | 0.186 | 90.001 | 176133 |
| 3 | 0 | 0.194 | 0.358 | 0.191 | 141.54 | 464040 |
| - | - | - | - | - | - | - |
| - | - | - | - | - | - | - |
| 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 |

Berdasarkan pada Tabel 4.1 akan dibuatkan plot untuk melihat pengaruh atribut lagu terhadap kinerja metode KNN dan MKNN melalui sebaran data lagu berdasarkan variabel fitur pada *genre* musik yang digunakan. Berikut adalah plot sebaran data dari variabel fitur yang digunakan:



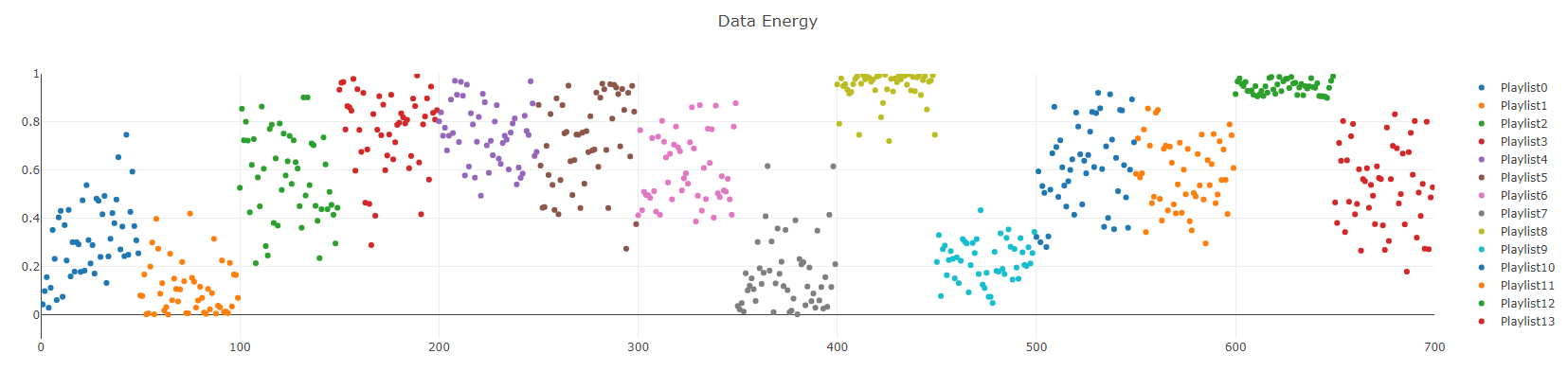
Gambar 4.15. Plot sebaran Danceability

Dilihat dari Gambar 4.15, terlihat sebaran data *danceability* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki perbedaan. Pada *genre* musik *EDM* sebaran data di 0.68 – 0.97, *genre* musik *Reggae* sebaran data di 0.64 – 0.94, *genre* musik *Soul* sebaran data di 0.61 – 0.94, *genre* musik *Funk* sebaran data di 0.53 – 0.89, sementara *genre* musik *R&B* memiliki sebaran data di 0.49 – 0.90, *genre* musik *Hip Hop* sebaran data di 0.38 – 0.96, *genre* musik *Blues* sebaran data di 0.29 – 0.77, *genre* musik *Pop* sebaran data di 0.27 – 0.75, *genre* musik *Country* sebaran data di 0.27 – 0.83, *genre* musik *Jazz* sebaran data di 0.21 – 0.82, *genre* musik *Gospel* sebaran data di 0.17 – 0.77, *genre* musik *Rock* sebaran data di 0.11 – 0.66, *genre* musik *Metal* sebaran data di 0.09 – 0.44, *genre* musik *Classical* memliki sebaran data di 0.07 – 0.60. Pada Gambar 4.15 menunjukkan bahwa musik *EDM*, *Reggae*, *Soul* dan *Funk* lebih cocok untuk menari dari pada musik *Blues*, *Classical*, *Country*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, dan *Rock*.



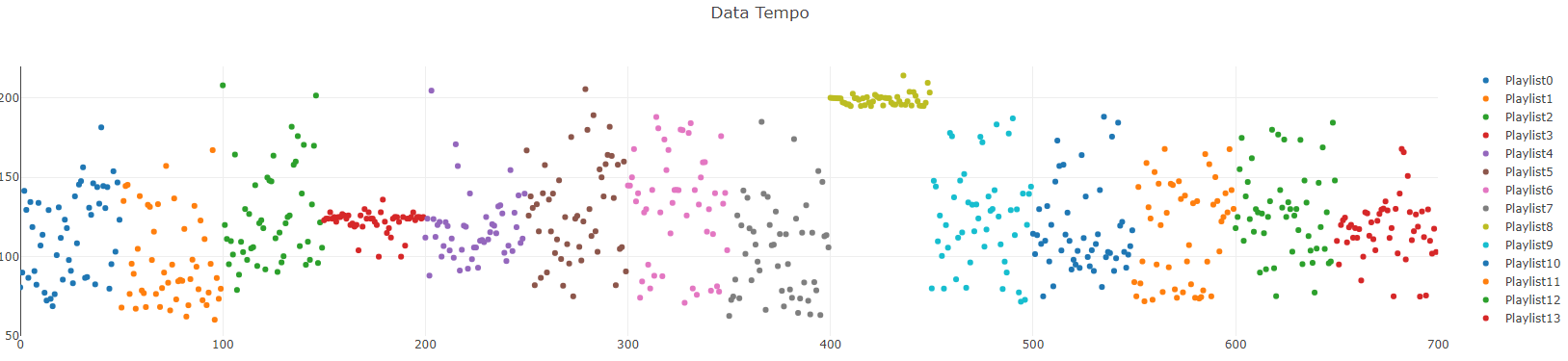
Gambar 4.16. Plot sebaran Valence

Dilihat dari Gambar 4.16, terlihat sebaran data *valence* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki perbedaan. Pada *genre* musik *Classical*, *Country*, *EDM*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki emosi yang bervariasi ketimbang musik *Blues* yang cenderung ke emosi negatif karena lebih banyak menyebar dibawah 0.5. Sedangkan *genre* musik *Funk* memiliki emosi yang gembira karena menyebar dari 0.9 – 1.0.



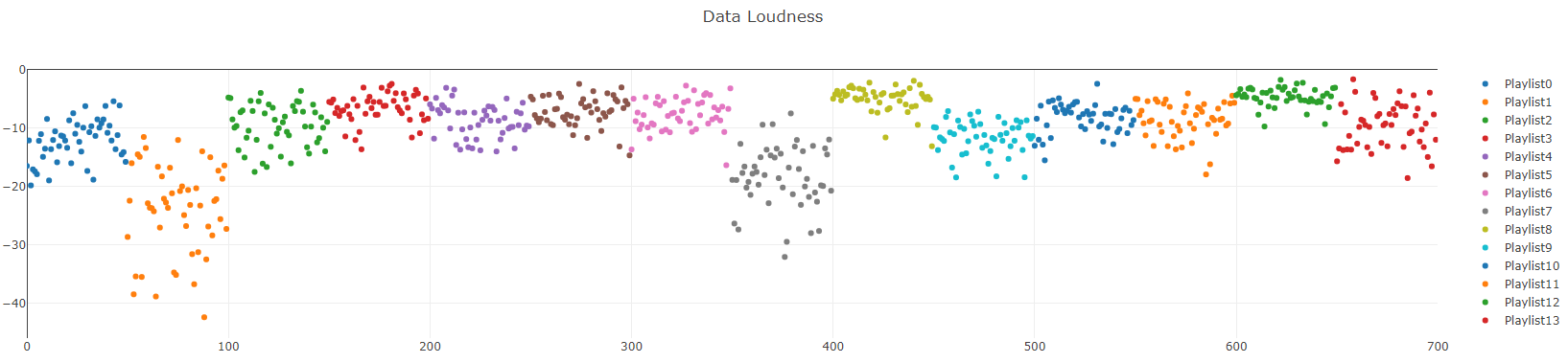
Gambar 4.17. Plot sebaran Energy

Dilihat dari Gambar 4.17, terlihat sebaran data *energy* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki perbedaan. Pada *genre* musik *Rock* sebaran data di 0.90 – 1.0, *genre* musik *Metal* sebaran data di 0.72 – 1.0, sementara *genre* musik *Funk* memiliki sebaran data di 0.49 – 0.98, *genre* musik *Hip Hop* sebaran data di 0.38 – 0.88, *genre* musik *Reggae* sebaran data di 0.29 – 0.86, *genre* musik *EDM* sebaran data di 0.28 – 1.0, *genre* musik *R&B* sebaran data di 0.28 – 0.93, *genre* musik *Gospel* sebaran data di 0.27 – 0.96, *genre* musik *Country* sebaran data di 0.21 – 0.90, *genre* musik *Soul* sebaran data di 0.17 – 0.84, *genre* musik *Pop* sebaran data di 0.04 – 0.44, *genre* musik *Blues* sebaran data di 0.02 – 0.75, *genre* musik *Jazz* sebaran data di 0.00 – 0.62, *genre* musik *Classical* sebaran data di 0.00 – 0.42. Pada Gambar 4.17 menunjukkan bahwa musik *Rock* dan *Metal* lebih energik dari pada musik *Blues*, *Classical*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Pop*, *R&B*, *Reggae*, dan *Soul*.



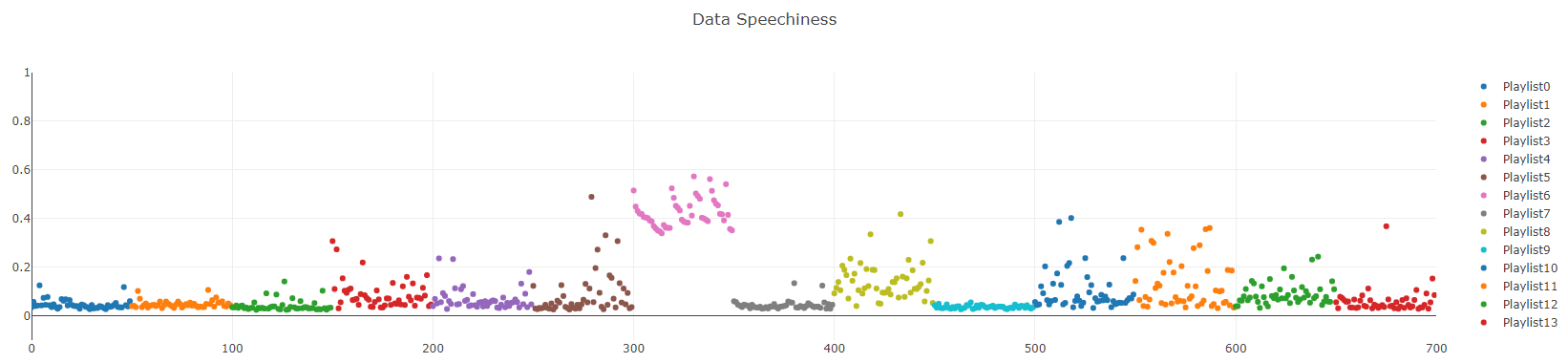
Gambar 4.18. Plot sebaran Tempo

Dilihat dari Gambar 4.18, terlihat sebaran data *tempo* pada *genre* musik *Blues*, *Country*, *Funk*, *Gospel*, *Hip Hop*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* secara umum hampir sama walaupun musik *Country*, *Funk* dan *Gospel* memiliki 1 data yang sebaran data nya tinggi di 200 bpm. Selanjutnya seluruh lagu pada *genre* musik *Metal* memiliki *tempo* tinggi yang menyebar di 194 bpm sampai 215 bpm. Sedangkan *genre* musik *EDM* menyebar rata di 100 bpm sampai 136 bpm. Pada Gambar 4.18 menunjukkan *genre* musik *Metal* memiliki *tempo* cepat dari pada *Blues*, *Country*, *EDM*, *Funk*, *Gospel*, *Hip Hop*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul*.



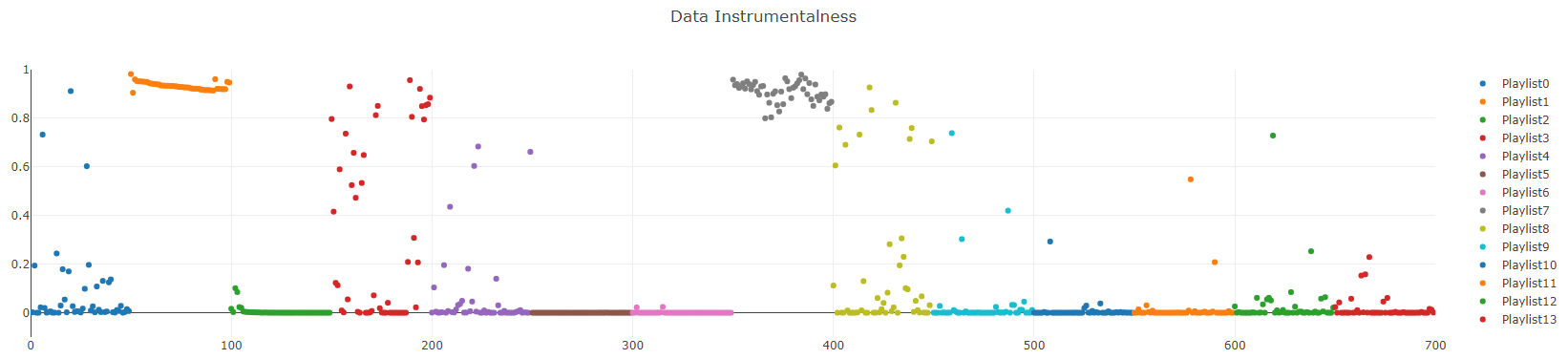
Gambar 4.19. Plot sebaran Loudness

Dilihat pada Gambar 4.19, terlihat sebaran data *loudness* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki perbedaan. Pada *genre* musik *Classical* sebaran data di -42.42 sampai -11.51, *genre* musik *Jazz* sebaran data di -32.08 sampai -7.47, sementara *genre* musik *Blues* memiliki sebaran data di -19.81 sampai -5.4, *genre* musik *Soul* sebaran data di -18.54 sampai -1.61, *genre* musik *Pop* sebaran data di -18.43 sampai -7.07, *genre* musik *Reggae* sebaran data di -17.92 sampai -4.03, *genre* musik *Country* sebaran data di -17.49 sampai -3.59, *genre* musik *Hip Hop* sebaran data di -16.34 sampai -2.70, *genre* musik *Gospel* sebaran data di -14.65 sampai -2.41, *genre* musik *Funk* sebaran data di -13.98 sampai -3.08, *genre* musik *EDM* sebaran data di -13.61 sampai -2.40, *genre* musik *Metal* sebaran data di -13.09 sampai -1.87, *genre* musik *Rock* sebaran data di -9.70 sampai -1.72. Pada Gambar 4.19 menunjukkan bahwa musik *Classical* dan *Jazz* lebih lembut dan menenangkan. dari pada musik *Blues*, *Country*, *Gospel*, *Hip Hop*, *Metal*, *Pop*, *Reggae*, *Rock*, *EDM,* *Funk*, *R&B*, dan *Soul*.



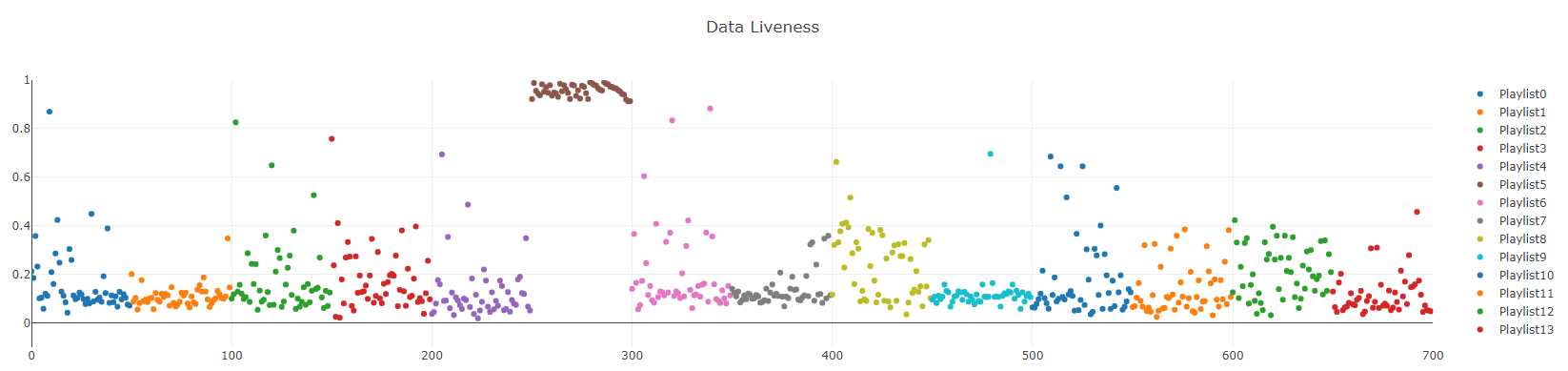
Gambar 4.20. Plot sebaran Speechiness

Dilihat dari Gambar 4.20, terlihat sebaran data *speechiness* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Jazz*, *Pop*, *Rock*, dan *Soul* memiliki sebaran data yang hampir sama yaitu rata–rata sebaran data dibawah 0.33, walaupun beberapa data dari musik *Gospel*, *Metal*, *R&B*, *Reggae*, dan *Soul* memiliki nilai *speechiness* lebih dari 0.33. Sedangkan *genre* musik *Hip Hop* memiliki nilai *speechiness* diantara 0.33 – 0.58. Pada Gambar 4.20 menunjukkan bahwa trek musik pada *genre* *Hip Hop* terdiri dari musik dan ucapan seperti musik lagu rap. Sementara nilai semakin dibawah 0.33 trek musik semakin minim ucapan atau kata-kata pada lagu tersebut.



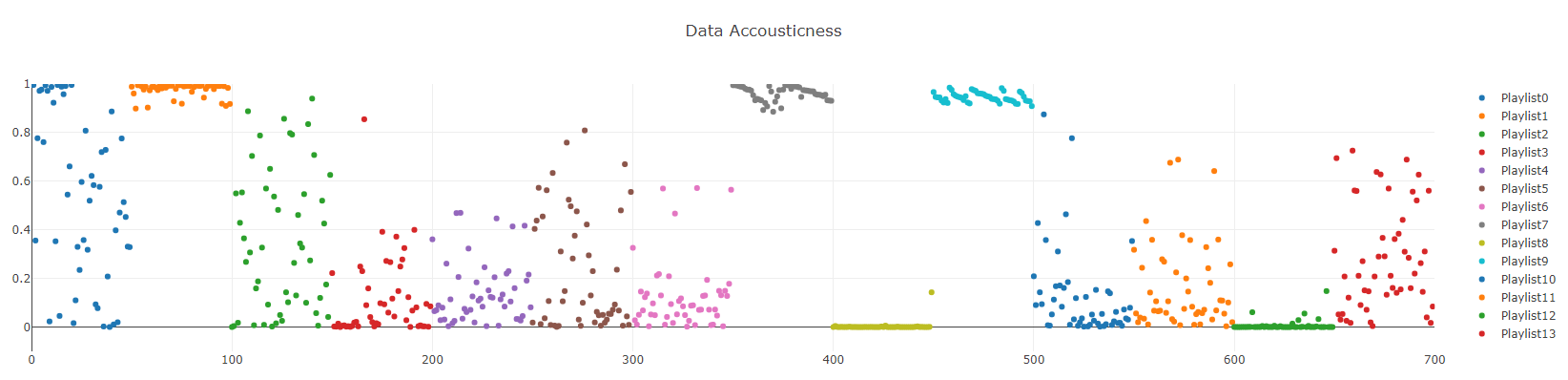
Gambar 4.21. Plot sebaran Instrumentalness

Dilihat pada Gambar 4.21, terlihat sebaran data *instrumentalness* pada *genre* musik *Blues*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki sebaran data yang hampir sama yaitu rata–rata sebaran data dibawah 0.30 walaupun beberapa data dari musik *Blues*, *EDM,* *Funk*, *Metal*, *Pop*, *Reggae*, *Rock*, memiliki beberapa nilai *instrumentalness* yang menyebar hingga 0.96. Sedangkan *genre* musik *Classical* memiliki nilai *instrumentalness* diantara 0.8 – 0.98, dan *genre* musik *Jazz* memiliki nilai *instrumentalness* diantara 0.90 – 0.99. Berdasarkan pengertian dari *instrumentalness* nilai semakin dekat ke nilai 1.0, maka semakin tinggi kecenderungan musik tersebut tidak disertai dengan vokal. Pada Gambar 4.21 menunjukkan bahwa trek musik pada *genre* *Classical* dan *Jazz* memiliki kecenderungan musik tidak disertai dengan vokal.



Gambar 4.22. Plot sebaran Liveness

Dilihat pada Gambar 4.22, terlihat sebaran data *liveness* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki sebaran data yang hampir sama yaitu rata–rata sebaran data dibawah 0.40 walaupun beberapa data dari musik *Blues*, *Country*, *EDM,* *Funk*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Rock*, dan *Soul*, memiliki nilai *liveness* yang menyebar hingga 0.9. Sedangkan *genre* musik *Gospel* memiliki nilai *liveness* diantara 0.91 sampai 1.0. Berdasarkan pengertian dari *liveness* nilai semakin dekat ke nilai 1.0, maka semakin tinggi kecenderungan musik tersebut melakukan rekaman saat *live* (siaran langsung). Pada Gambar 4.22 menunjukkan bahwa trek musik pada *genre* *Gospel* melakukan rekaman lagu saat *live* (siaran langsung).



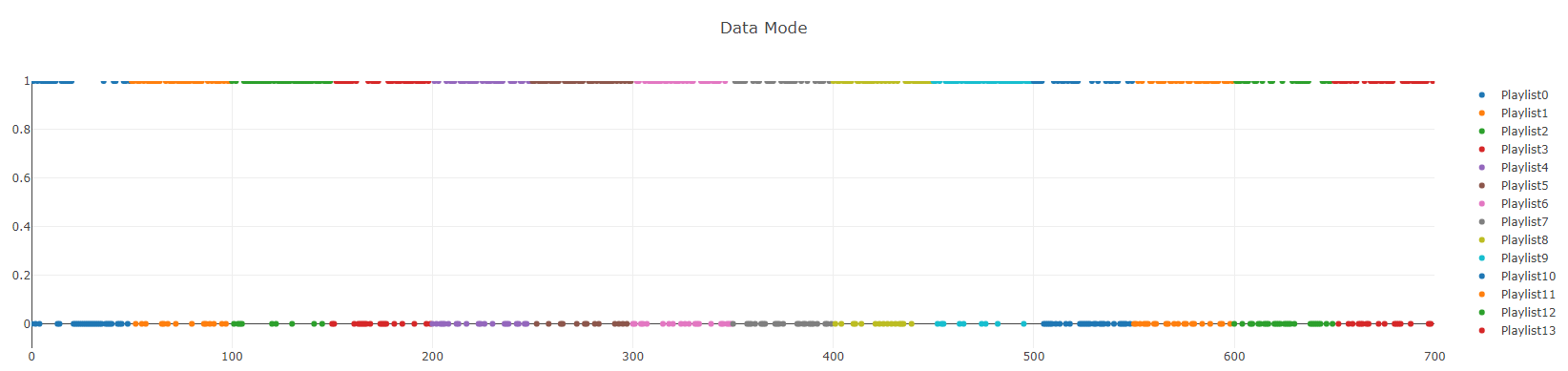
Gambar 4.23. Plot sebaran Accousticness

Dilihat pada Gambar 4.23, terlihat sebaran data *accousticness* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*,dan *Soul* memiliki perbedaan. Pada *genre* musik *Pop* sebaran data di 0.90 – 0.99, *genre* musik *Classical* sebaran data di 0.89 – 1.0, *genre* musik *Jazz* sebaran data di 0.88 – 1.0, sementara *genre* musik *Soul* memiliki sebaran data di 0.00 – 0.73, *genre* musik *Funk* sebaran data di 0.00 – 0.47, *genre* musik *Reggae* sebaran data di 0.00 – 0.69, *genre* musik *R&B* sebaran data di 0.00 – 0.88, *genre* musik *Hip Hop* sebaran data di 0.00 – 0.58, *genre* musik *Gospel* sebaran data di 0.00 – 0.81, *genre* musik *Blues* sebaran data di 0.00 – 1.0, *genre* musik *Country* sebaran data di 0.00 – 0.94, *genre* musik *EDM* sebaran data di 0.00 – 0.86, *genre* musik *Rock* sebaran data di 0.00 – 0.15, *genre* musik *Metal* sebaran data di 0.00 – 0.15. Pada Gambar 4.23 menunjukkan bahwa musik *Classical*, *Jazz* dan *Pop* lebih banyak menggunakan trek alat musik akustik dari pada musik *Blues*, *Country*, *Gospel*, *Hip Hop*, *Metal*, *Reggae*, *Rock*, *EDM,* *Funk*, *R&B*, dan *Soul*.



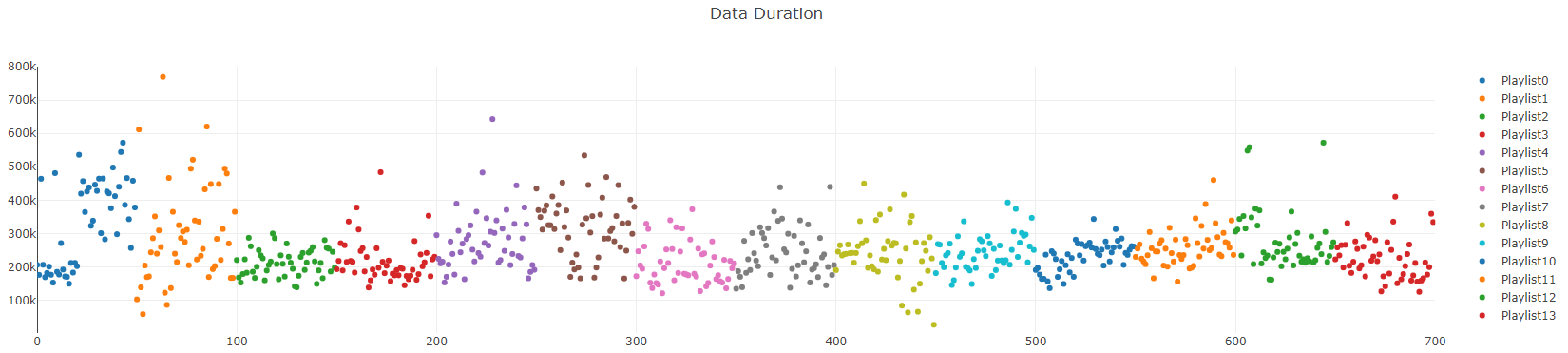
Gambar 4.24. Plot sebaran Key

Dilihat pada Gambar 4.24, terlihat sebaran data *Key* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki sebaran data yang merata dari 0 hingga 11. Pada Gambar 4.24 menunjukkan bahwa 14 *genre* musik tersebut menggunakan semua trek kunci tangga nada.



Gambar 4.25. Plot sebaran Mode

Dilihat dari Gambar 4.25, terlihat bahwa sebaran data dari variabel fitur *Mode* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* memiliki sebaran data yang merata dari 0 dan 1. Pada Gambar 4.25 menunjukkan bahwa 14 *genre* musik tersebut menggunakan semua trek kunci melodi minor dan mayor. Berdasarkan pengertian dari *mode*, tangga nada Mayor bersifat Menggembirakan, sedangkan Minor bersifat Menyedihkan. Mayor bernada dasar C=do, sedangkan Minor bernada dasar A=la.



Gambar 4.26. Plot sebaran Duration

Dilihat pada Gambar 4.26, terlihat sebaran data *duration* pada *genre* musik *Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* secara umum hampir sama memiliki berbagai macam durasi panjang nya lagu walaupun beberapa data dari musik *Classical* dan *Funk* memiliki durasi panjang lebih dari 600000 milidetik atau 10 menit.

## **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 tranformasi 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 tranformasi 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\_ms |
| 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

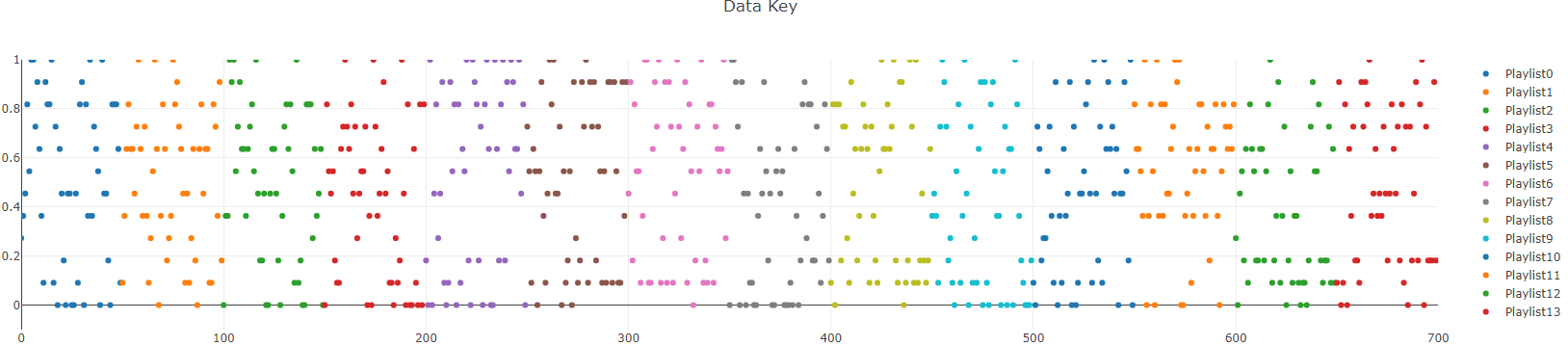
= 0.635983

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.

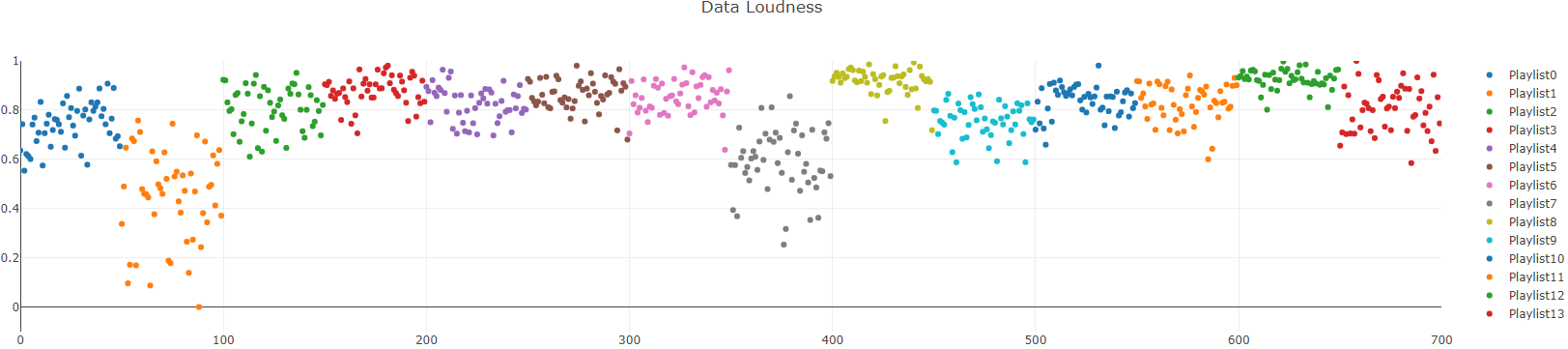
Tabel 4.3. Data Variabel Fitur setelah ditransformasi

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

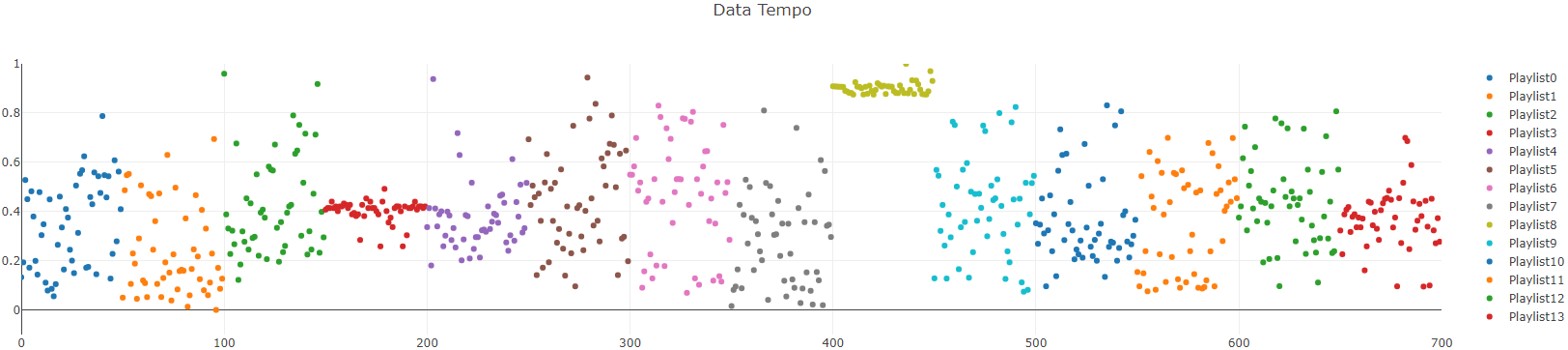
Jika dilihat melalui sebaran data maka *range* data variabel fitur *key, loudness, tempo,* dan *duration* seperti gambar dibawah ini:



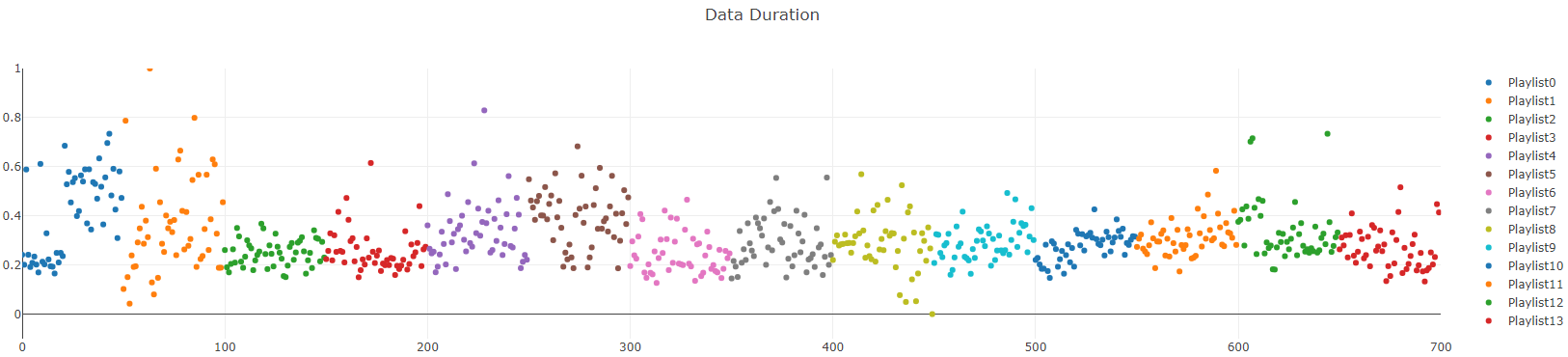
Gambar 4.27. Sebaran Key setelah ditransformasi



Gambar 4.28. Sebaran Loudness setelah ditransformasi



Gambar 4.29. Sebaran Tempo setelah ditransformasi



Gambar 4.30. Sebaran *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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Playlist | Danceability | Energy | Key | Loudness | Mode | Speechiness | Accousticness |
| 1 | 0 | 0.538 | 0.0466 | 0.272727 | 0.635983 | 0 | 0.0349 | 0.996 |
| 2 | 0 | 0.599 | 0.0424 | 0.363636 | 0.74298 | 1 | 0.0572 | 0.996 |
| 3 | 0 | 0.402 | 0.0982 | 0.454545 | 0.553952 | 0 | 0.0401 | 0.356 |
| - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - |
| 698 | 13 | 0.745 | 0.272 | 0.181818 | 0.634194 | 0 | 0.0551 | 0.561 |
| 699 | 13 | 0.939 | 0.487 | 0.909091 | 0.852208 | 0 | 0.153 | 0.0172 |
| 700 | 13 | 0.921 | 0.529 | 0.181818 | 0.746043 | 1 | 0.0851 | 0.0845 |

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

## **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.



### **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 ditujukkan pada baris ke 3 sampai dengan 164 pada Lampiran bagian B.2 untuk pengujian pertama, sedangkan pada pengujian kedua pada Lampiran bagian C.2 baris ke 3 sampai dengan 164. 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 165 sampai dengan 189.

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 191 sampai dengan 206 pada Lampiran bagian C.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 ditujukkan 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 3 sampai dengan 164. 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 165 sampai dengan 189.

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.

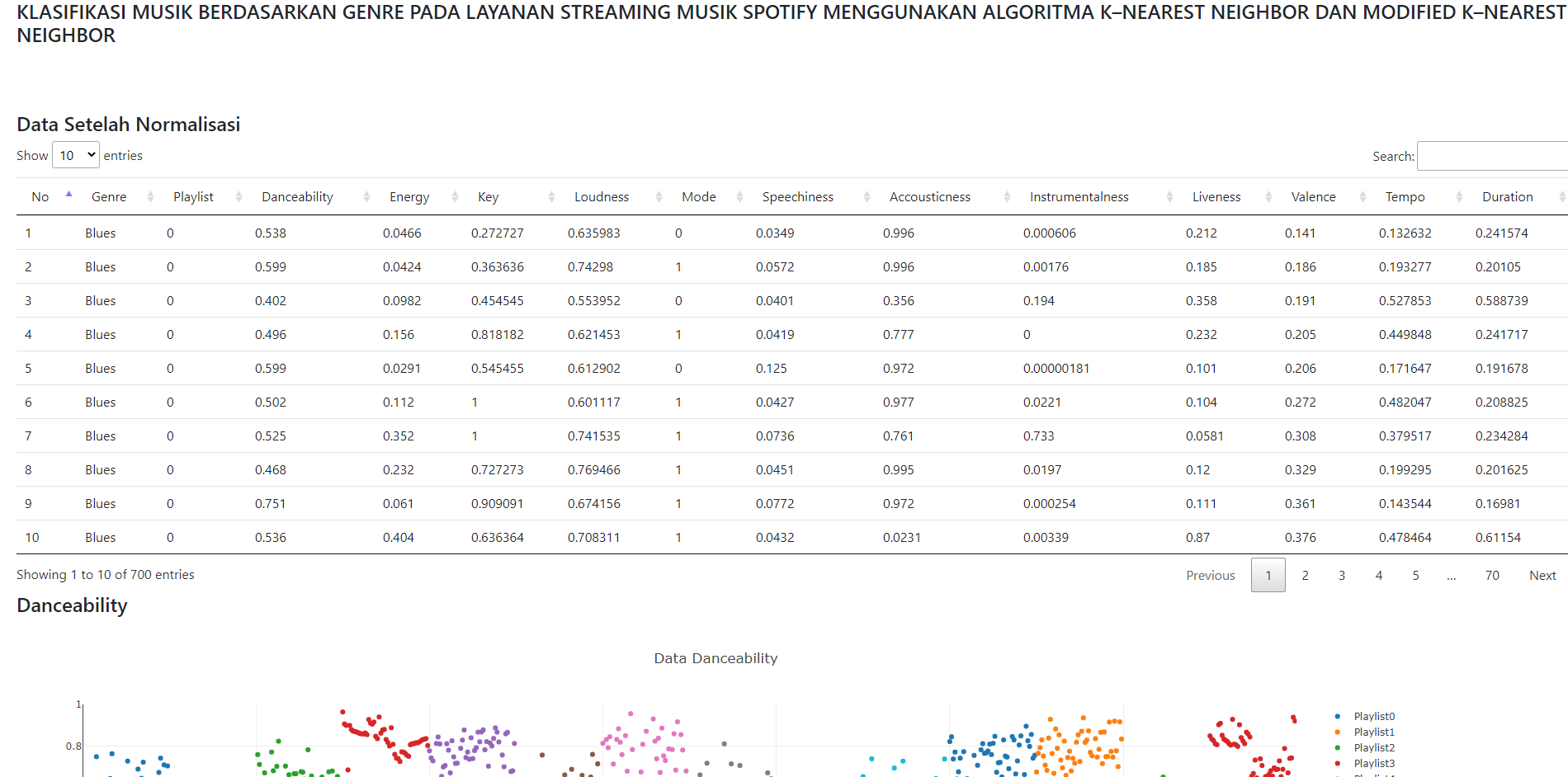
Sedangkan pengujian kedua, proses Persamaan (2.10) dan Persamaan (2.11) tersebut berada pada baris ke 191 sampai dengan 229 pada Lampiran bagian C.3. Pembobotan terdapat pada baris ke 230 sampai dengan 237 pada Lampiran bagian C.3. Selanjutnya kelas data dengan bobot tertinggi menjadi hasil kelas data 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 238 sampai dengan 250 pada Lampiran bagian C.3.

## **Implementasi Sistem**

Pengguna melakukan *min–max normalization* yang nanti nya 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 PCAKNN dan PCAMKNN untuk Pengujian Kedua.

## **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.



### **Pengujian Pertama**



Pada pengujian pertama ini akan membandingkan akurasi dari tiap perubahan nilai *k* pada *K–Nearest Neighbor* (KNN) dan *Modified K–Nearest Neighbor* (MKNN) menggunakan metode *k–fold cross–validation* dengan nilai *k* adalah 10. Data yang digunakan sebanyak 700 data lagu, dengan pembagian jumlah data yaitu 90% data *training* (data latih) sebanyak 630 data lagu dan 10% data *testing* (data uji) sebanyak 70 data lagu. Untuk 70 lagu data *testing* (data uji) akan diambil secara merata yaitu 5 data lagu pada tiap–tiap *genre* atau tiap kelas data. Sehingga, data dipartisi menjadi 10 bagian dengan masing–masing partisi berisi 70 data. Sebelum melakukan klasifikasi pada 700 data lagu akan di acak terlebih dahulu, agar tidak sesuai urutan tiap *genre* nya. Pada Tabel 4.5 disajikan tampilan data yang sudah diacak.

Tabel 4.5. Pengacakkan Data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Playlist | Danceability | Energy | Key | Loudness | Mode | Speechiness | Acousticness |
| 1 | 2 | 0.657 | 0.415 | 0.454545 | 0.796491 | 1 | 0.025 | 0.175 |
| 2 | 1 | 0.166 | 0.0171 | 0.363636 | 0.458274 | 1 | 0.0396 | 0.985 |
| 3 | 4 | 0.816 | 0.603 | 0.636364 | 0.747268 | 1 | 0.0749 | 0.134 |
| - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - |
| 698 | 6 | 0.839 | 0.432 | 0 | 0.790366 | 0 | 0.492 | 0.572 |
| 699 | 3 | 0.837 | 0.81 | 0 | 0.919488 | 1 | 0.0395 | 0.00136 |
| 700 | 1 | 0.208 | 0.15 | 0.636364 | 0.591978 | 1 | 0.0585 | 0.985 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Playlist | Instrumentalness | Liveness | Valence | Tempo | Duration\_ms |
| 1 | 2 | 0.00000565 | 0.132 | 0.347 | 0.232416 | 0.214804 |
| 2 | 1 | 0.941 | 0.123 | 0.0361 | 0.0519465 | 0.313409 |
| 3 | 4 | 0.0000427 | 0.152 | 0.963 | 0.35405 | 0.463309 |
| - | - | - | - | - | - | - |
| - | - | - | - | - | - | - |
| 698 | 6 | 0.00000915 | 0.117 | 0.496 | 0.128127 | 0.135562 |
| 699 | 3 | 0.858 | 0.256 | 0.805 | 0.414131 | 0.263596 |
| 700 | 1 | 0.934 | 0.119 | 0.546 | 0.105633 | 0.147792 |



#### **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. Nilai *k* yang digunakan pada penelitian ini yaitu 2 sampai dengan 29 karena pemilihan nilai *k* tersebut berdasarkan hasil akurasi yang didapatkan. Pada Gambar 4.33 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 5 nilai rata–rata akurasi cenderung mengalami kenaikan, selanjutnya hasil akurasi mengalami fluktuasi saat *k* bernilai 6 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* = 5, dimana nilai akurasi yang diperoleh yaitu 62%. Nilai akurasi terendah terjadi pada saat nilai *k* = 29, dimana nilai akurasi yang diperoleh yaitu 52.7%. Selanjutnya nilai *k* = 4 dan *k* = 8 memiliki nilai akurasi yang sama yaitu 60.4%, untuk nilai *k* = 11 dan *k* = 12 memiliki nilai akurasi yang sama yaitu 59%, untuk nilai *k* = 14 dan *k* = 15 memiliki nilai akurasi yang sama yaitu 58.9%, sedangkan untuk nilai *k* = 23 dan *k* = 24 memiliki nilai akurasi yang sama yaitu 54.9%.

Gambar 4.33. Akurasi MKNN

#### **Perbandingan Akurasi pada perubahan nilai k MKNN dengan KNN**

Dari keseluruhan nilai *k* pada data uji klasifikasi *genre*, metode MKNN memiliki akurasi yang lebih rendah dibandingkan dengan metode KNN. Pada nilai *k* = 5 dengan nilai akurasi 62% merupakan akurasi tertinggi pada metode MKNN, sedangkan pada metode KNN memiliki nilai akurasi sebesar 63%. Sehingga ketika nilai *k* = 5 (*k* yang paling optimal), metode KNN meningkatkan nilai akurasi dari metode MKNN pada saat nilai *k* = 5 sebanyak 1.0%.

Selanjutnya pada saat nilai *k* = 7 dengan nilai akurasi 63.3% merupakan akurasi tertinggi pada metode KNN, sedangkan pada metode MKNN cenderung menurun dengan nilai akurasi menjadi 61%. Sehingga ketika nilai *k* = 7 (*k* yang paling optimal), metode KNN meningkatkan nilai akurasi dari metode MKNN pada saat nilai *k* = 7 sebanyak 2.3%. 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*

#### **Perbandingan Akurasi 10–Fold Cross–Validation MKNN dengan KNN pada k = 5**

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* = 5. Pada saat nilai *k–fold* iterasike*–*3 dengan nilai akurasi 70% merupakan akurasi tertinggi pada metode MKNN, sedangkan saat nilai *k–fold* iterasi ke*–*3 dengan nilai akurasi 70% merupakan akurasi tertinggi pada metode KNN. Pada Gambar 4.35 disajikan hasil perbandingan akurasi 10*–fold cross–validation* MKNN dengan KNN pada *k* = 5.

Gambar 4.35. Perbandingan Akurasi 10*–Fold* *Cross*–*Validation* MKNN dengan KNN pada *k* = 5

#### **Perbandingan Akurasi 10–Fold Cross–Validation MKNN dengan KNN pada k = 7**

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* = 7. Pada saat nilai *k–fold* iterasike*–*8 dengan nilai akurasi 68.6% merupakan akurasi tertinggi pada metode MKNN, sedangkan saat nilai *k–fold* iterasi ke*–*8 dengan nilai akurasi 71.4% merupakan akurasi tertinggi pada metode KNN. Pada Gambar 4.36 disajikan hasil perbandingan akurasi 10*–fold cross–validation* MKNN dengan KNN pada *k* = 7.

Gambar 4.36. Perbandingan Akurasi 10*–Fold* *Cross*–*Validation* MKNN dengan KNN pada *k* = 7

#### **Akurasi Klasifikasi Genre Lagu menggunakan KNN pada Confusion Matrix saat nilai k = 5 di k–fold iterasi ke–3**

Akurasi 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 Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3:

Tabel 4.6. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 40% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 40% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 80% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 20% |

Gambar 4.37. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3

Pada Gambar 4.37 menunjukkan jumlah kelas pada *playlist* label 1 *Classical,* label 5 *Gospel*, label 8 *Metal* dan label 9 *Pop* mendapatkan akurasi sebesar 100% karena tiap lima data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan untuk *playlist* label lainnya mendapatkan akurasi dibawah 100%, karena ada beberapa data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian. Nilai akurasi terendah sebesar 20% karena empat data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian terjadi pada *playlist* label 2 *Country* dan label 13 *Soul*.

Selanjutnya pada *playlist* label 3 *EDM,* label 4 *Funk*, label 11 *Reggae* dan label 12 *Rock* mendapatkan akurasi yang sama sebesar 80% karena tiap empat data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, selanjutnya pada *playlist* label 0 *Blues,* label 6 *Hip Hop* dan label 7 *Jazz* mendapatkan akurasi yang sama sebesar 60% karena tiap tiga data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan pada *playlist* label 10 *R&B* mendapatkan akurasi sebesar 40% karena tiap dua data uji *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian. Untuk tabel *confusion matrix* lainnya bisa dilihat pada Lampiran D.4.

#### **Akurasi Klasifikasi Genre Lagu menggunakan MKNN pada Confusion Matrix saat nilai k = 5 di k–fold iterasi ke–3**

Akurasi klasifikasi *genre* lagu menggunakan metode MKNN pada *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 Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3:

Tabel 4.7 Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 40% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 40% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 80% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 20% |

Gambar 4.38. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*3

Pada Gambar 4.38 menunjukkan jumlah kelas pada *playlist* label 1 *classical*, label 5 *Gospel* dan label 9 *Pop* mendapatkan akurasisebesar 100% karena tiap lima data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan untuk *playlist* label lainnya mendapatkan akurasi dibawah 100%, karena ada beberapa data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian. Nilai akurasi terendah sebesar 20% karena empat data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian terjadi pada *playlist* label 2 *Country* dan label 13 *Soul*.

Selanjutnya pada *playlist* label 3 *EDM,* label 4 *Funk*, label 7 *Jazz*, label 8 *Metal*, label 11 *Reggae* dan label 12 *Rock* mendapatkan akurasi yang sama sebesar 80% karena tiap empat data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, selanjutnya pada *playlist* label 0 *Blues* danlabel 6 *Hip Hop* mendapatkan akurasi yang sama sebesar 60% karena tiap tiga data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan pada *playlist* label 10 *R&B* mendapatkan akurasi yang sama sebesar 40% karena tiap dua data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian. Untuk Tabel *confusion matrix* lainnya bisa dilihat pada Lampiran D.5.

#### **Akurasi Klasifikasi Genre Lagu menggunakan KNN pada Confusion Matrix saat nilai k = 7 di k–fold iterasi ke–8**

Akurasi 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 4.8 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8:

Tabel 4.8. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
| t | 3 | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 80% | 0% |
|  | 13 | 20% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% |

Gambar 4.39. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8

Pada Gambar 4.39 menunjukkan jumlah kelas pada *playlist* label 3 *EDM*, label 4 *Funk*, label 5 *Gospel*, label 8 *Metal* dan label 9 *Pop* mendapatkan akurasi sebesar 100% karena tiap lima data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan untuk *playlist* label lainnya mendapatkan akurasi dibawah 100%, karena ada beberapa data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian. Nilai akurasi terendah sebesar 0% karena lima data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian terjadi pada *playlist* label 13 *Soul*.

Selanjutnya pada *playlist* label 1 *Classical,* label 6 *Hip Hop* dan label 12 *Rock* mendapatkan akurasi yang sama sebesar 80% karena tiap empat data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, selanjutnya pada *playlist* label 0 *Blues*,label 2 *Country*, label 7 *Jazz* dan label 11 *Reggae* mendapatkan akurasi yang sama sebesar 60% karena tiap tiga data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan pada *playlist* label 10 *R&B* mendapatkan akurasi sebesar 20% karena tiap satu data uji dari *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian. Untuk tabel *confusion matrix* lainnya bisa dilihat pada Lampiran D.6.

#### **Akurasi Klasifikasi Genre Lagu menggunakan MKNN pada Confusion Matrix saat nilai k = 7 di k–fold iterasi ke–8**

Akurasi klasifikasi *genre* lagu menggunakan metode MKNN pada *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 4.9 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8:

Tabel 4.9 Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
| t | 3 | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 11 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% |

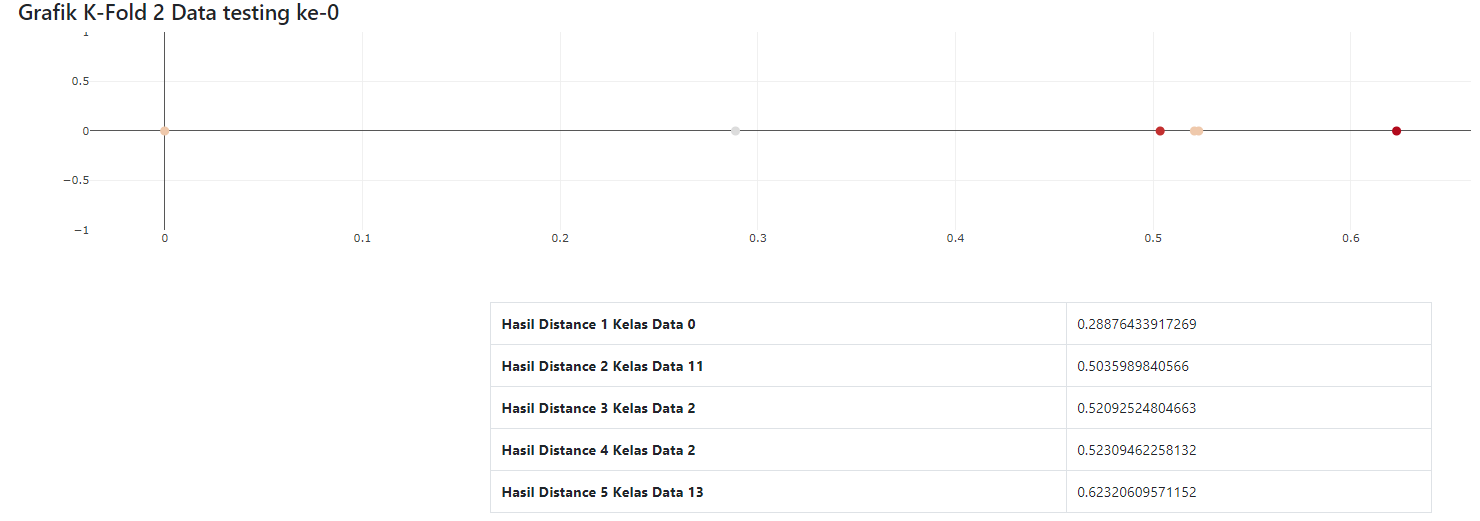
Gambar 4.40. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*8

Pada Gambar 4.40 menunjukkan jumlah kelas pada *playlist* label 4 *Funk*, label 5 *Gospel*, label 8 *Metal*, label 9 *Pop* dan label 12 *Rock* mendapatkan akurasisebesar 100% karena tiap lima data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan untuk *playlist* label lainnya mendapatkan akurasi dibawah 100%, karena ada beberapa data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian. Nilai akurasi terendah sebesar 0% karena lima data uji yang tidak dapat menebak kelas sesuai dengan label yang diberikan dalam pengujian terjadi pada *playlist* label 10 *Reggae* dan label 13 *Soul*.

Selanjutnya pada *playlist* label 3 *EDM* dan label 6 *Hip Hop* mendapatkan akurasi yang sama sebesar 80% karena tiap empat data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian, sedangkan pada *playlist* label 0 *Blues*, label 1 *Classical*,label 2 *Country*, label 7 *Jazz* dan label 11 *Reggae* mendapatkan akurasi yang sama sebesar 60% karena tiap tiga data uji dari masing*–*masing *genre* tersebut dapat menebak kelas benar sesuai dengan label yang diberikan dalam pengujian. Untuk Tabel *confusion matrix* lainnya bisa dilihat pada Lampiran D.7.

#### **Identifikasi Pemilihan Kelas pada KNN saat nilai k = 5 di k–fold iterasi ke–3**

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* = 5 di *k–fold* iterasike*–*3 sebagai berikut:



Gambar 4.41. Perhitungan *euclidean distance* KNN sejumlah *k* =5 pada data testing ke-0

Pada Gambar 4.41 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* = 5 maka dipilih *euclidean* *distance* sejumlah 5 nilai *euclidean* *distance*. Diperoleh hasil sebagai berikut:

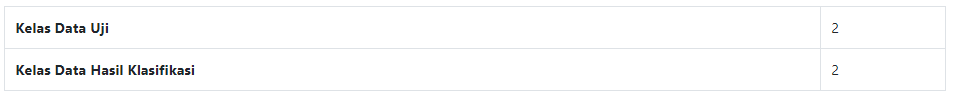
**Hasil Distance 1 Kelas Data 0 :** 0.28876433917269

**Hasil Distance 2 Kelas Data 11 :** 0.5035989840566

**Hasil Distance 3 Kelas Data 2 :** 0.52092524804663

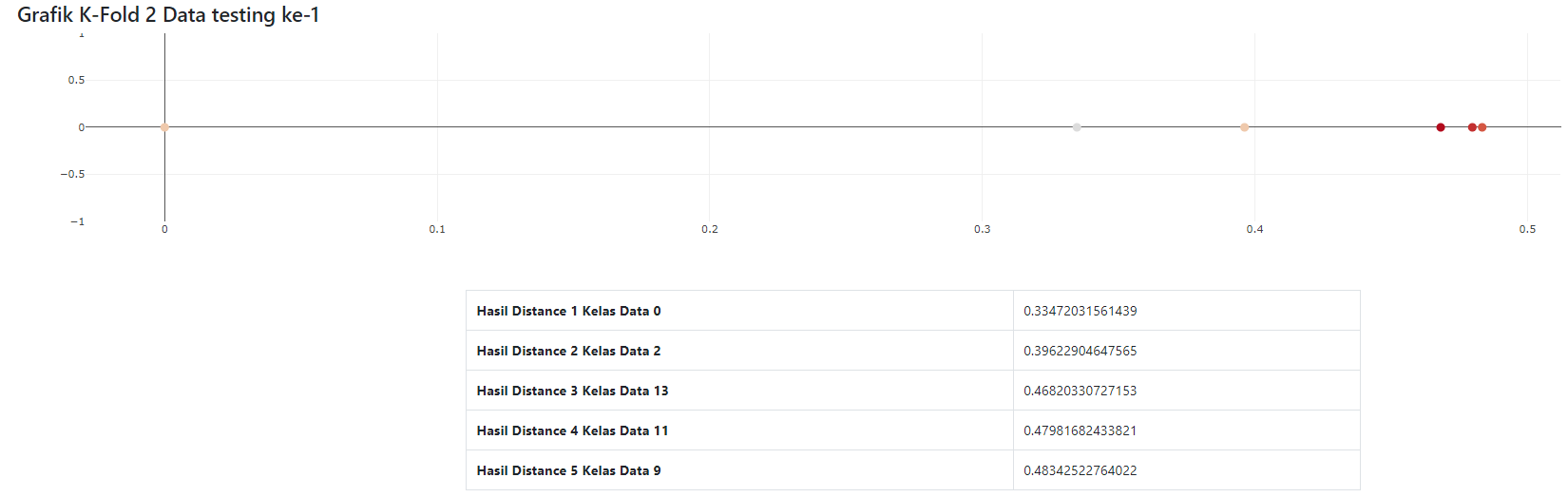
**Hasil Distance 4 Kelas Data 2 :** 0.52309462258132

**Hasil Distance 5 Kelas Data 13 :** 0.62320609571152



Gambar 4.42. Kelas Data Hasil Klasifikasi 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 4.42 menunjukkan kelas label 2 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* = 5 di *k–fold* iterasike*–*3 pada data testing ke-0 yang terpilih adalah label 2 yaitu benar, sedangkan pemilihan label *genre* yang salah pada klasifikasi KNN sebagai berikut:



Gambar 4.43. Perhitungan *euclidean distance* KNN sejumlah *k* =5 pada data testing ke-1

Dilihat pada Gambar 4.43 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* = 5 maka dipilih *euclidean* *distance* sejumlah 5 nilai *euclidean* *distance*. Diperoleh hasil sebagai berikut:

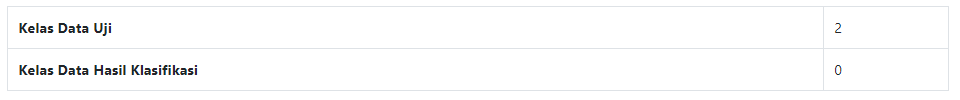
**Hasil Distance 1 Kelas Data 0 :** 0.33472031561439

**Hasil Distance 2 Kelas Data 2 :** 0.39622904647565

**Hasil Distance 3 Kelas Data 13 :** 0.46820330727153

**Hasil Distance 4 Kelas Data 11 :** 0.47981682433821

**Hasil Distance 5 Kelas Data 9 :** 0.48342522764022



Gambar 4.44. Kelas Data Hasil Klasifikasi 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 4.44 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* = 5 di *k–fold* iterasike*–*3 pada data testing ke-1 yang terpilih adalah label 0 yaitu salah.

#### **Identifikasi Pemilihan Kelas pada MKNN saat nilai k = 5 di k–fold iterasi ke–3**

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* = 5 di *k–fold* iterasike*–*3 sebagai berikut:



Gambar 4.45. Perhitungan *euclidean distance* MKNN sejumlah *k* =5 pada data testing ke-0

Pada Gambar 4.45 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* = 5 maka dipilih sejumlah 5 nilai *euclidean* *distance*. Diperoleh hasil sebagai berikut:

**Hasil Distance 1 Kelas Data 0 :** 0.28876433917269

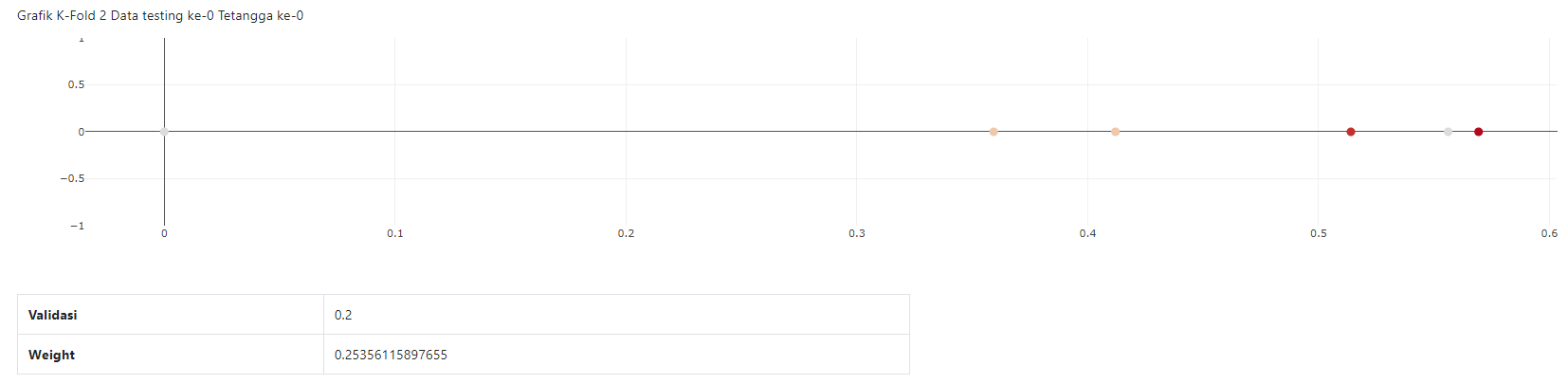
**Hasil Distance 2 Kelas Data 11 :** 0.5035989840566

**Hasil Distance 3 Kelas Data 2 :** 0.52092524804663

**Hasil Distance 4 Kelas Data 2 :** 0.52309462258132

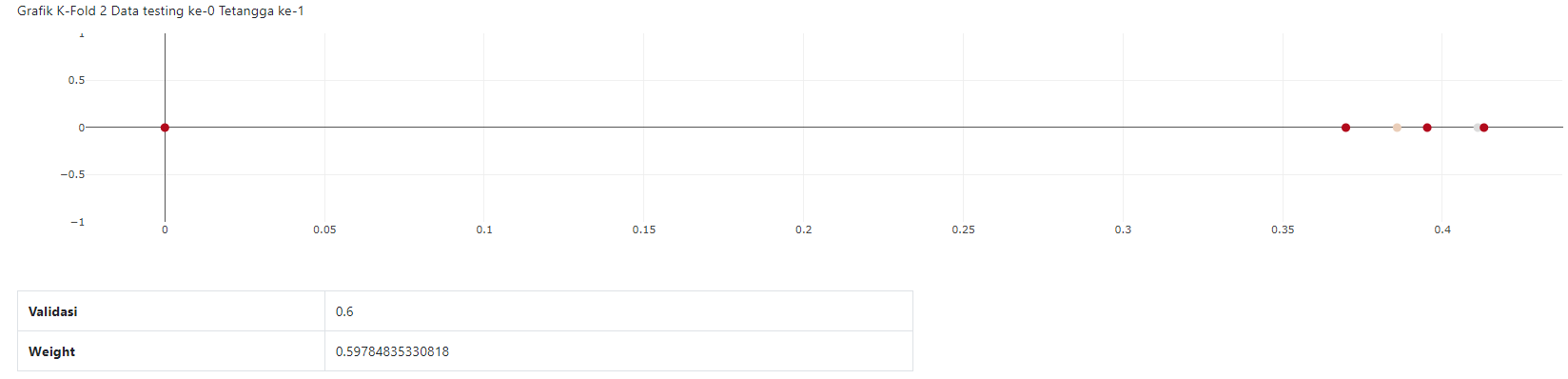
**Hasil Distance 5 Kelas Data 13 :** 0.62320609571152

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 tetangga nya maka bernilai 1, tetapi jika tidak sama dengan data latih tetangga nya 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.



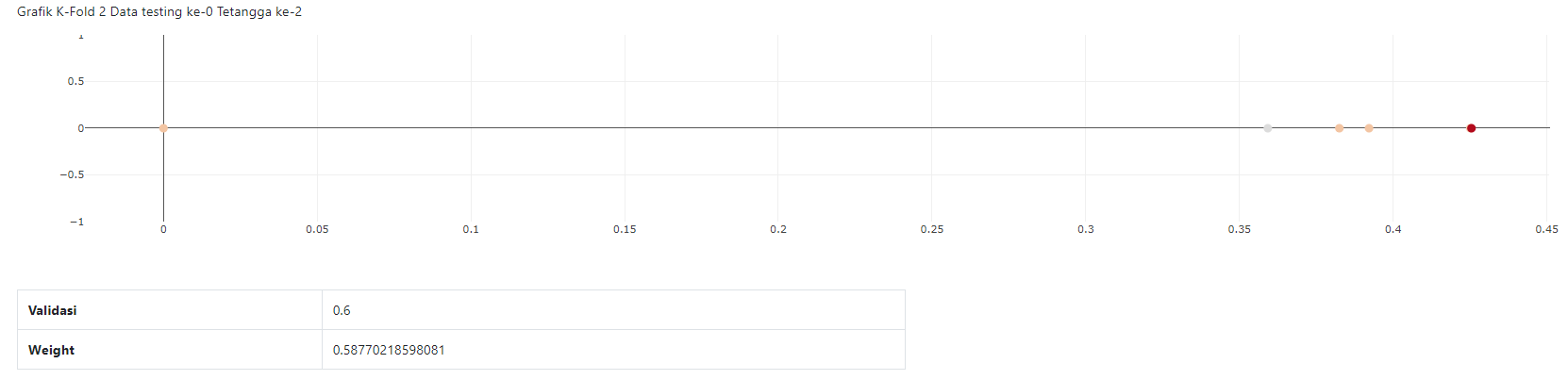
Gambar 4.46. Menghitung validasi dan bobot data testing ke-0 tetangga ke-0

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 4.46 yaitu nilai validasi adalah 0.2 dan nilai bobot adalah 0.25356115897655.



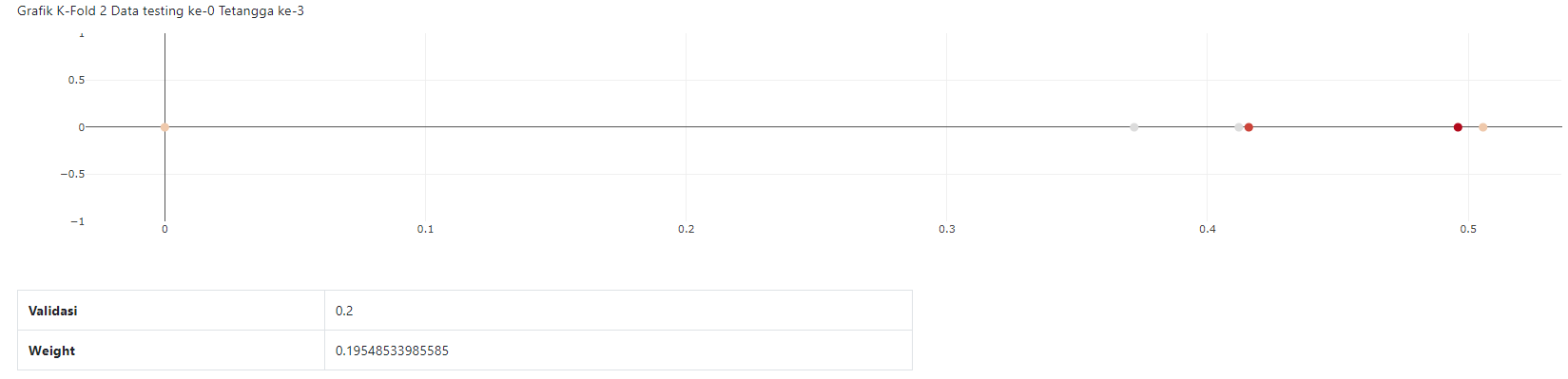
Gambar 4.47. Menghitung validasi dan bobot data testing ke-0 tetangga ke-1

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 4.47 yaitu nilai validasi adalah 0.6 dan nilai bobot adalah 0.59784835330818.



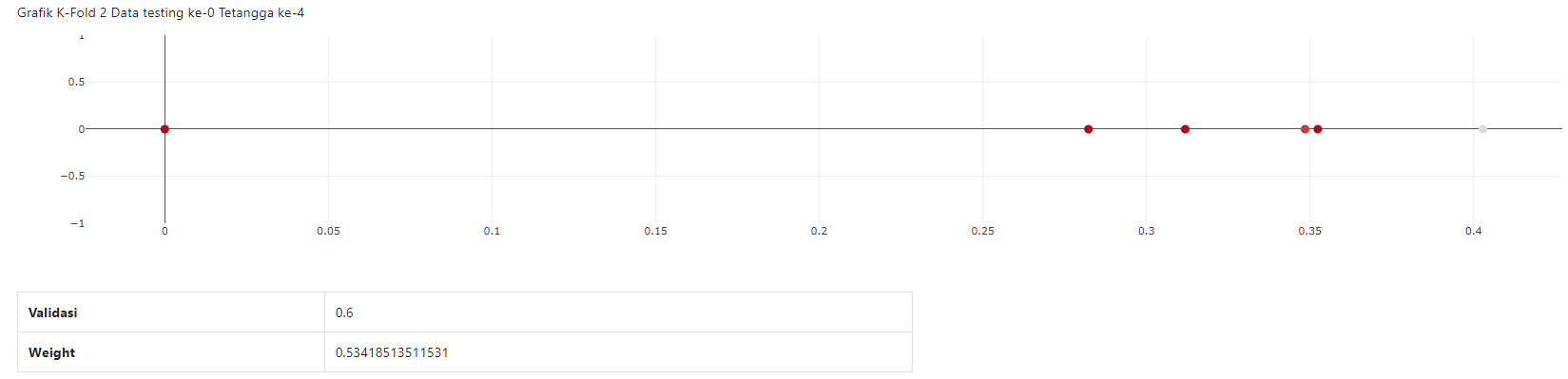
Gambar 4.48. Menghitung validasi dan bobot data testing ke-0 tetangga ke-2

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 4.48 yaitu nilai validasi adalah 0.6 dan nilai bobot adalah 0.58770218598081.



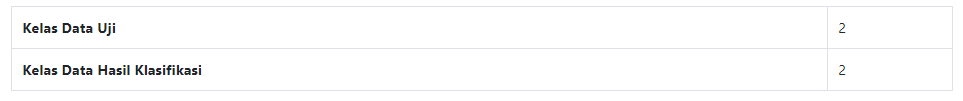
Gambar 4.49. Menghitung validasi dan bobot data testing ke-0 tetangga ke-3

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 4.49 yaitu nilai validasi adalah 0.2 dan nilai bobot adalah 0.19548533985585.



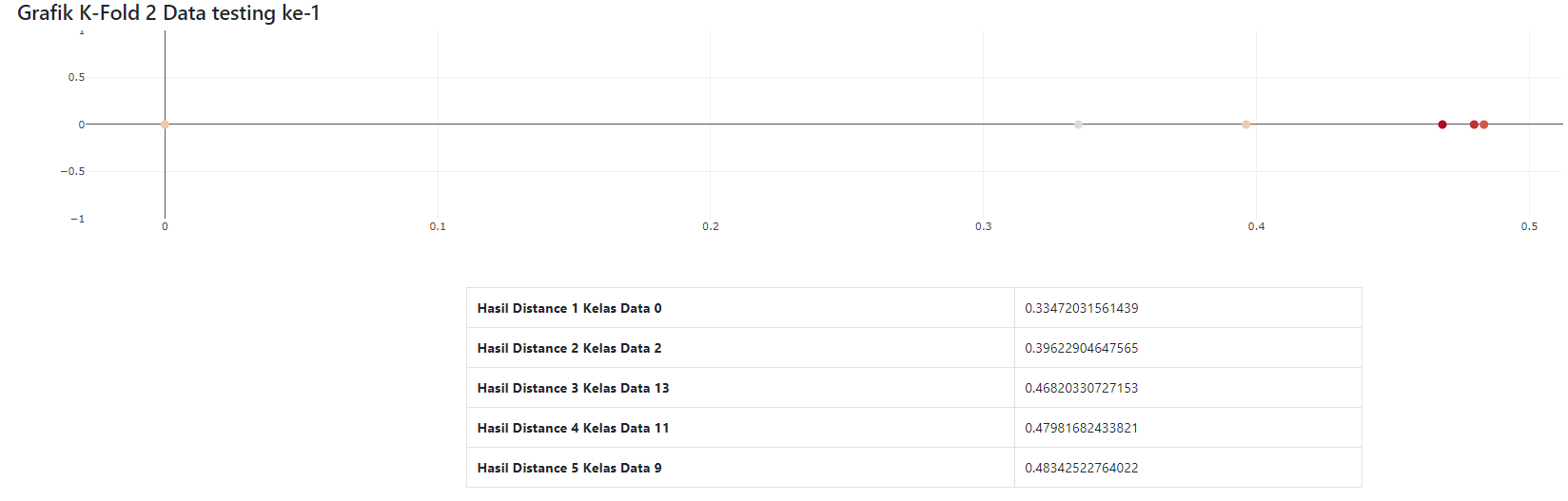
Gambar 4.50. Menghitung validasi dan bobot data testing ke-0 tetangga ke-4

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 4.50 yaitu nilai validasi adalah 0.6 dan nilai bobot adalah 0.53418513511531.



Gambar 4.51. Kelas Data Hasil Klasifikasi yang terpilih benar

Dari rumus Persamaan (2.12) pada Gambar 4.51 menunjukkan kelas label 2 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* = 5 dan *k–fold* iterasike*–*3 dan data testing ke-0 menghasilkan kelas data hasil klasifikasi yang terpilih adalah label 2 yaitu benar, sedangkan pemilihan label *genre* yang salah pada MKNN sebagai berikut:



Gambar 4.52. Perhitungan *euclidean distance* MKNN sejumlah *k* =5 pada data testing ke-1

Dilihat pada Gambar 4.52, 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* = 5 hasil *distance* terpilih sejumlah *k* = 5 sebagai berikut:

**Hasil Distance 1 Kelas Data 0 :** 0.33472031561439

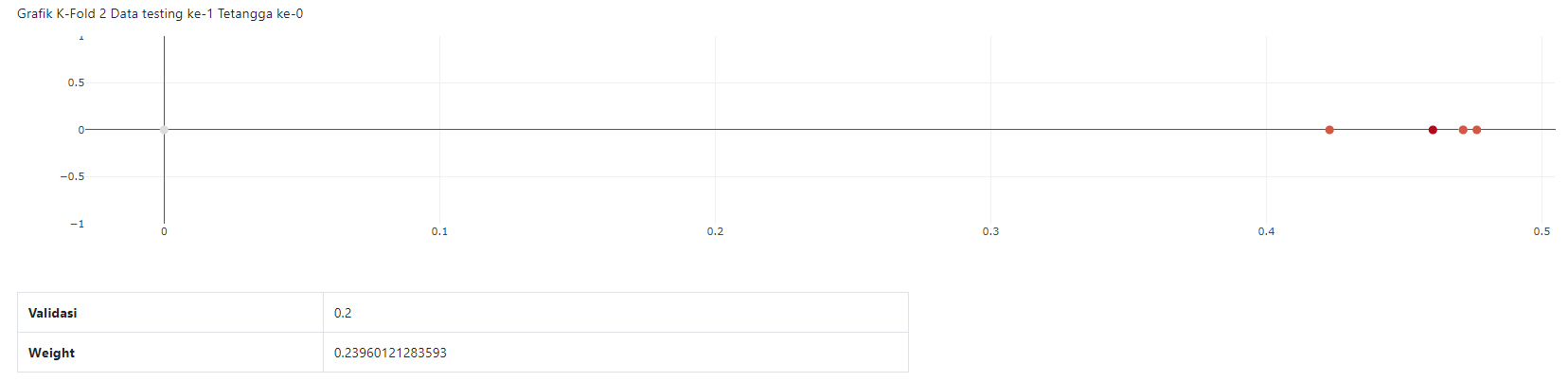
**Hasil Distance 2 Kelas Data 2 :** 0.39622904647565

**Hasil Distance 3 Kelas Data 13 :** 0.46820330727153

**Hasil Distance 4 Kelas Data 11 :** 0.47981682433821

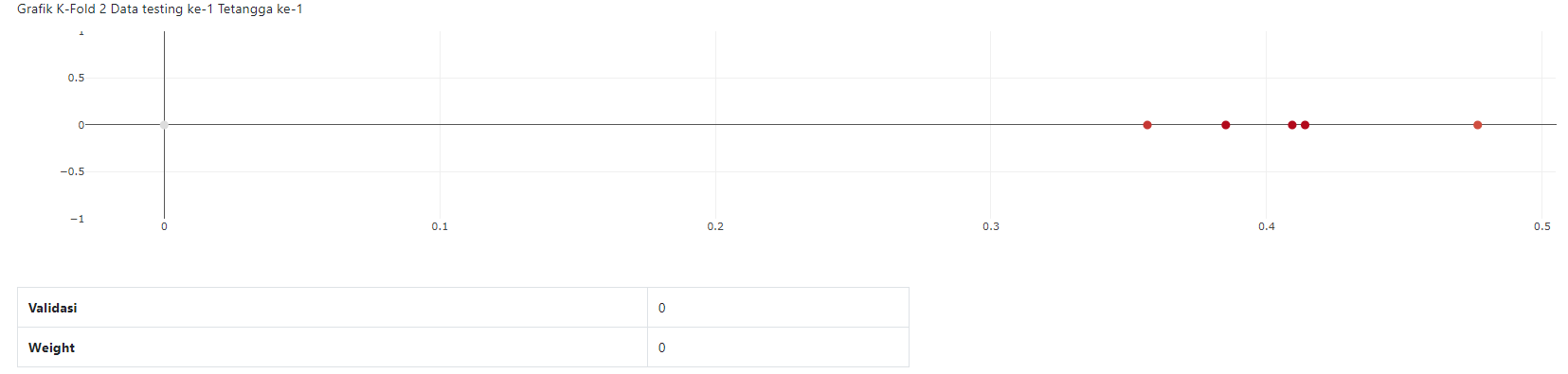
**Hasil Distance 5 Kelas Data 9 :** 0.48342522764022

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 tetangga nya maka bernilai 1, tetapi jika tidak sama dengan data latih tetangga nya 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.



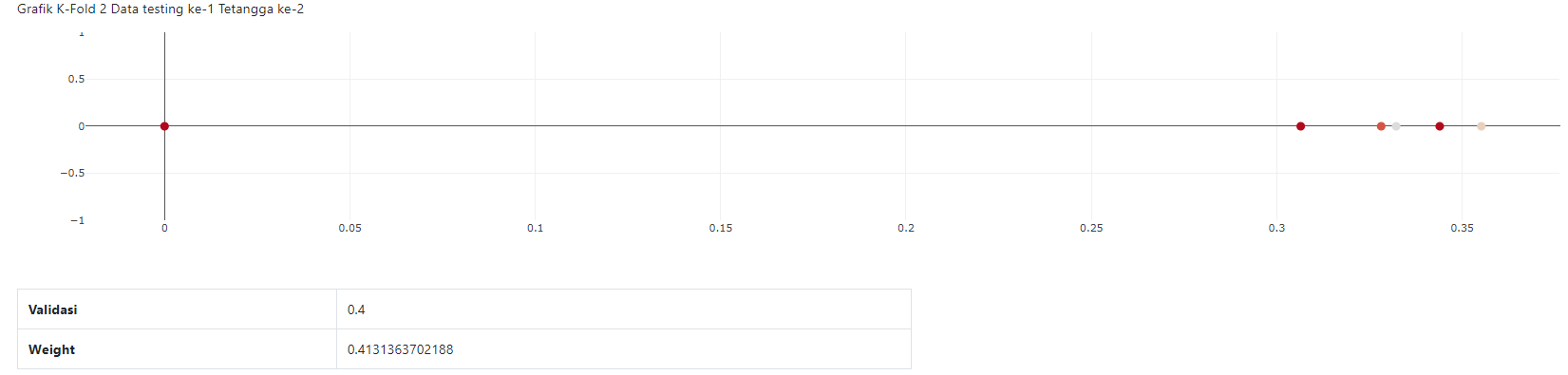
Gambar 4.53. Menghitung validasi dan bobot data testing ke-1 tetangga ke-0

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-1 dengan tetangga ke-0 pada Gambar 4.53 yaitu nilai validasi yang didapatkan adalah 0.2 dan nilai bobot yang didapatkan adalah 0.23960121283593.



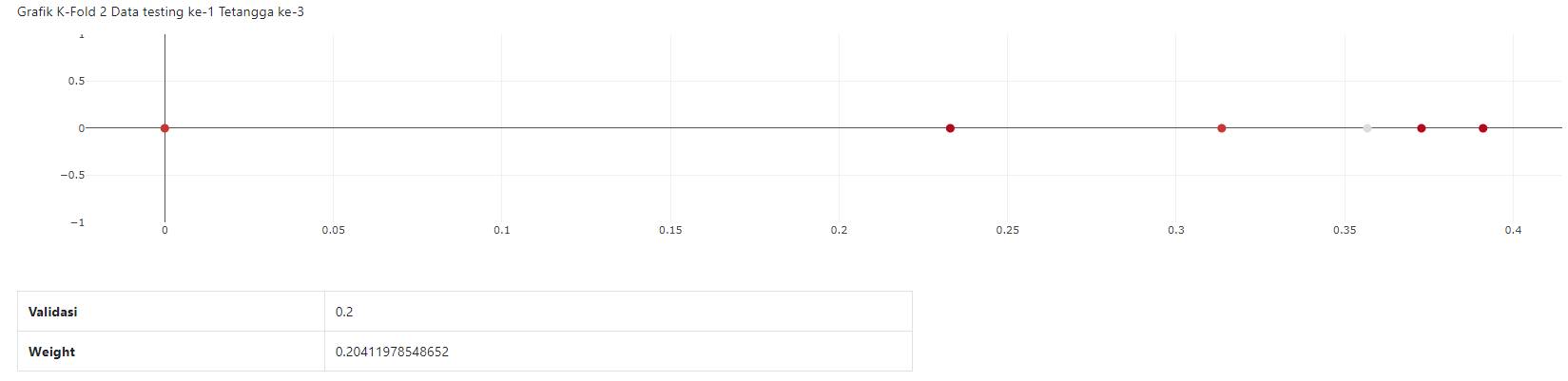
Gambar 4.54. Menghitung validasi dan bobot data testing ke-1 tetangga ke-1

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-1 dengan tetangga ke-1 pada Gambar 4.54 yaitu nilai validasi yang didapatkan adalah 0 dan nilai bobot yang didapatkan adalah 0.



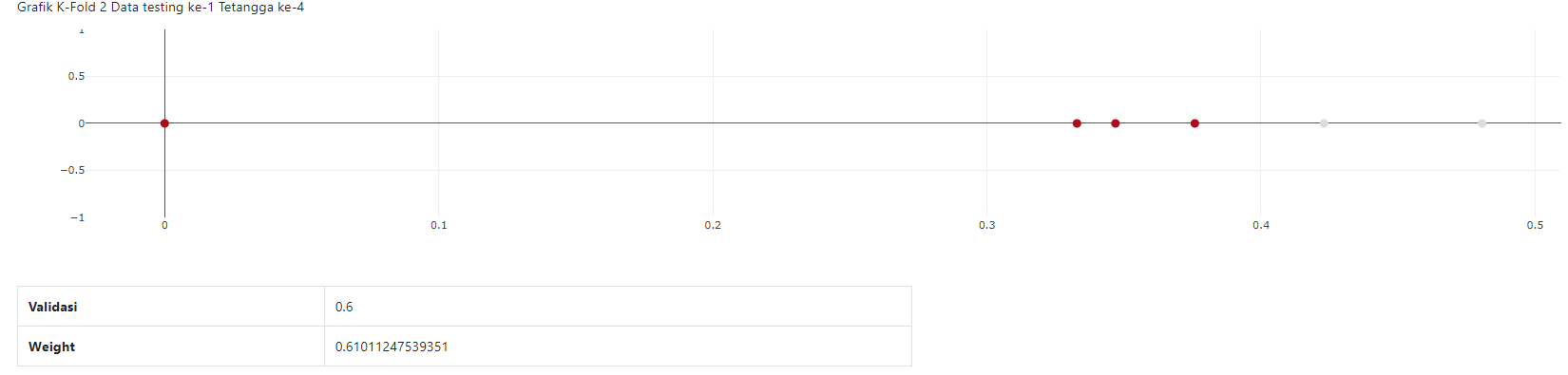
Gambar 4.55. Menghitung validasi dan bobot data testing ke-1 tetangga ke-2

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-1 dengan tetangga ke-2 pada Gambar 4.55 yaitu nilai validasi yang didapatkan adalah 0.4 dan bobot yang didapatkan adalah 0.4131363702188.



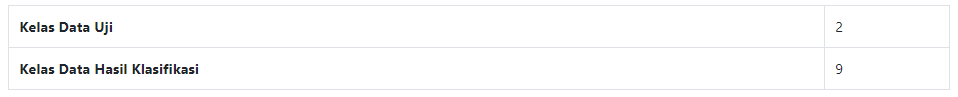
Gambar 4.56. Menghitung validasi dan bobot data testing ke-1 tetangga ke-3

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-1 dengan tetangga ke-3 pada Gambar 4.56 yaitu nilai validasi yang didapatkan adalah 0.2 dan yang didapatkan adalah 0.20411978548652.



Gambar 4.57. Menghitung validasi dan bobot data testing ke-1 tetangga ke-4

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-1 dengan tetangga ke-4 pada Gambar 4.57 yaitu nilai validasi yang didapatkan adalah 0.6 dan bobot yang didapatkan adalah 0.61011247539351.

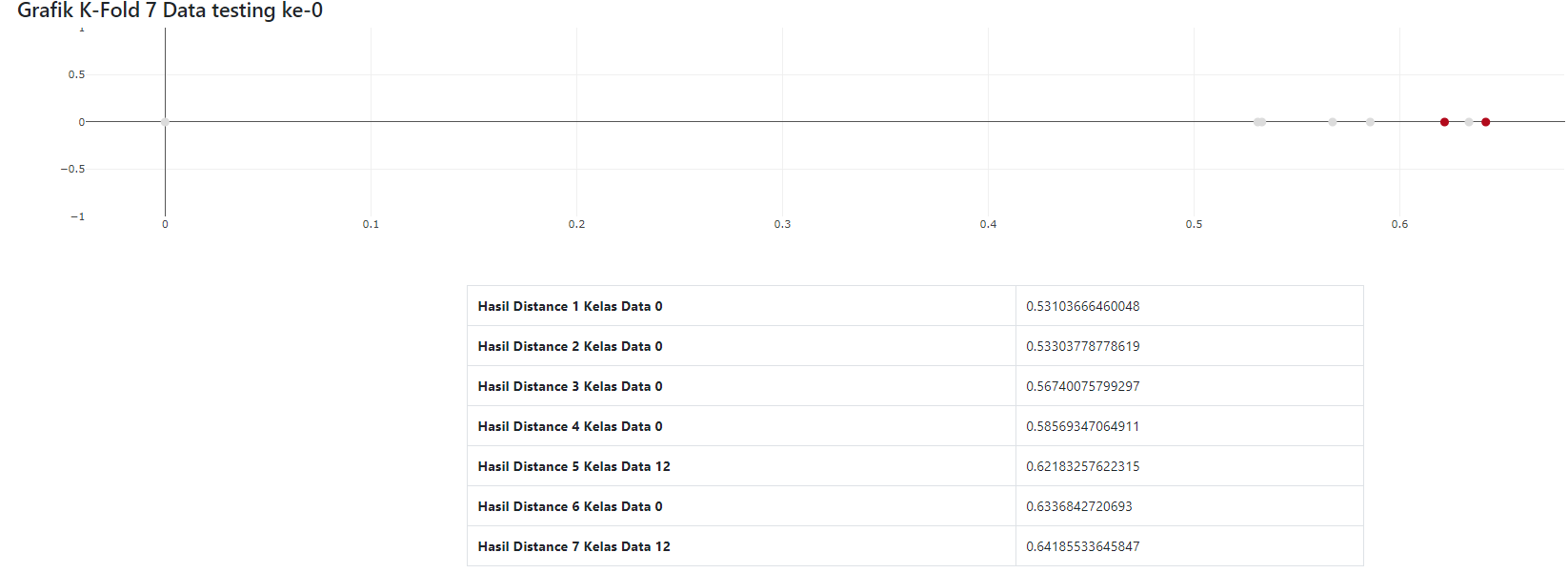


Gambar 4.58. Kelas Data Hasil Klasifikasi yang terpilih salah

Dari rumus Persamaan (2.12) pada Gambar 4.58 menunjukkan kelas label 2 yang sebagai kelas data uji pada data testing ke-1 memilih bobot yang tertinggi yang akan terpilih sebagai kelas data hasil klasifikasi. Pada klasifikasi MKNN saat nilai *k* = 5 dan *k–fold* iterasike*–*3 dan data testing ke-1 menghasilkan kelas data hasil klasifikasi yang terpilih adalah label 0 yaitu salah.

#### **Identifikasi Pemilihan Kelas pada KNN saat nilai k = 7 di k–fold iterasi ke–8**

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 4.8. Diperoleh hasil pemilihan kelas yang benar diklasifikasi maupun yang salah diklasifikasi pada metode KNN saat nilai *k* = 7 di *k–fold* iterasike*–*8 sebagai berikut:



Gambar 4.59. Perhitungan *euclidean distance* KNN sejumlah *k* = 7 pada data testing ke-0

Pada Gambar 4.59 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* = 7 maka dipilih *euclidean* *distance* sejumlah 7 nilai *euclidean* *distance*. Diperoleh hasil sebagai berikut:

**Hasil Distance 1 Kelas Data 0 :** 0.53103666460048

**Hasil Distance 2 Kelas Data 0 :** 0.53303778778619

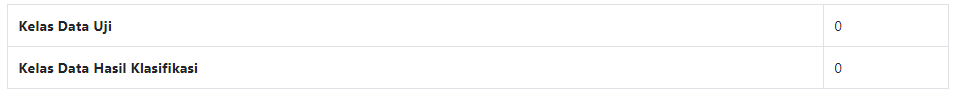
**Hasil Distance 3 Kelas Data 0 :** 0.56740075799297

**Hasil Distance 4 Kelas Data 0 :** 0.58569347064911

**Hasil Distance 5 Kelas Data 12 :** 0.62183257622315

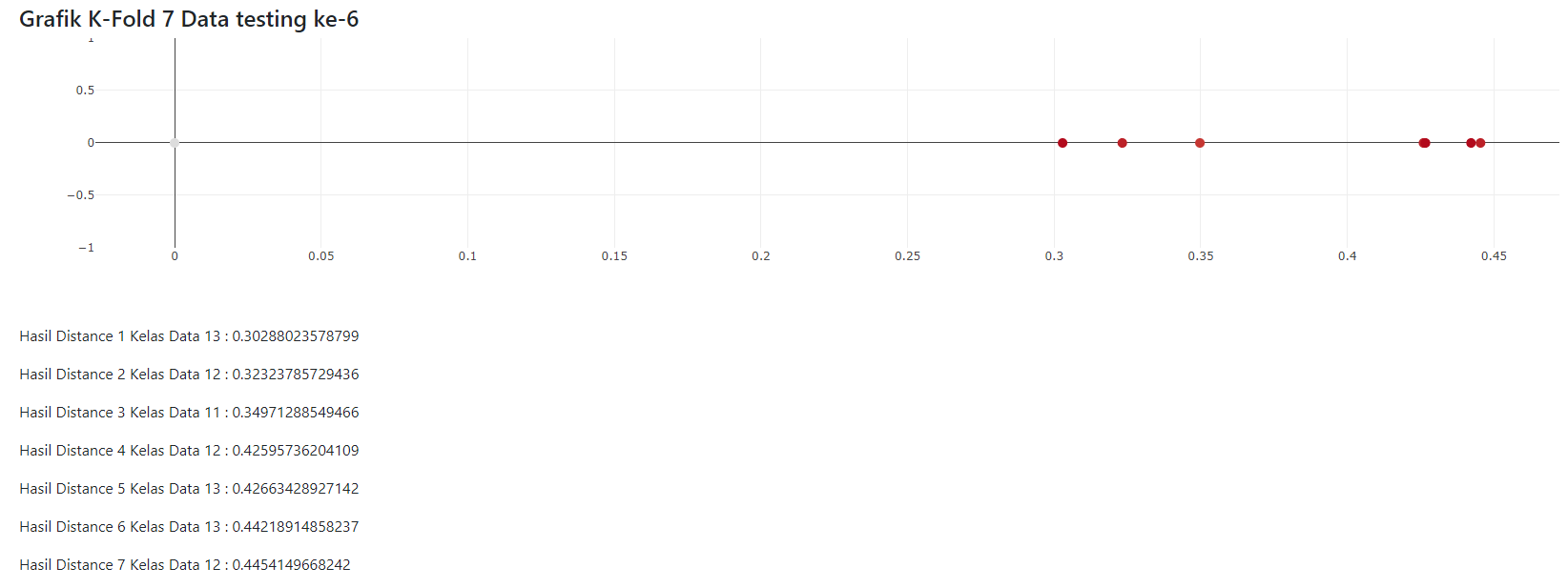
**Hasil Distance 6 Kelas Data 0 :** 0.6336842720693

**Hasil Distance 7 Kelas Data 12 :** 0.64185533645847



Gambar 4.60. Kelas Data Hasil Klasifikasi 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 4.60 menunjukkan kelas label 0 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* = 7 di *k–fold* iterasike*–*8 pada data testing ke-0 yang terpilih adalah label 0 yaitu benar, sedangkan pemilihan label *genre* yang salah pada klasifikasi KNN sebagai berikut:



Gambar 4.61. Perhitungan *euclidean distance* KNN sejumlah *k* = 7 pada data testing ke-6

Dilihat pada Gambar 4.61 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* = 7 maka dipilih *euclidean* *distance* sejumlah 7 nilai *euclidean* *distance*. Diperoleh hasil sebagai berikut:

**Hasil Distance 1 Kelas Data 13 :** 0.30288023578799

**Hasil Distance 2 Kelas Data 12 :** 0.32323785729436

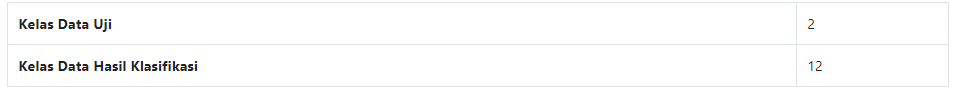
**Hasil Distance 3 Kelas Data 11 :** 0.34971288549466

**Hasil Distance 4 Kelas Data 12 :** 0.42595736204109

**Hasil Distance 5 Kelas Data 13 :** 0.42663428927142

**Hasil Distance 6 Kelas Data 13 :** 0.44218914858237

**Hasil Distance 7 Kelas Data 12 :** 0.4454149668242

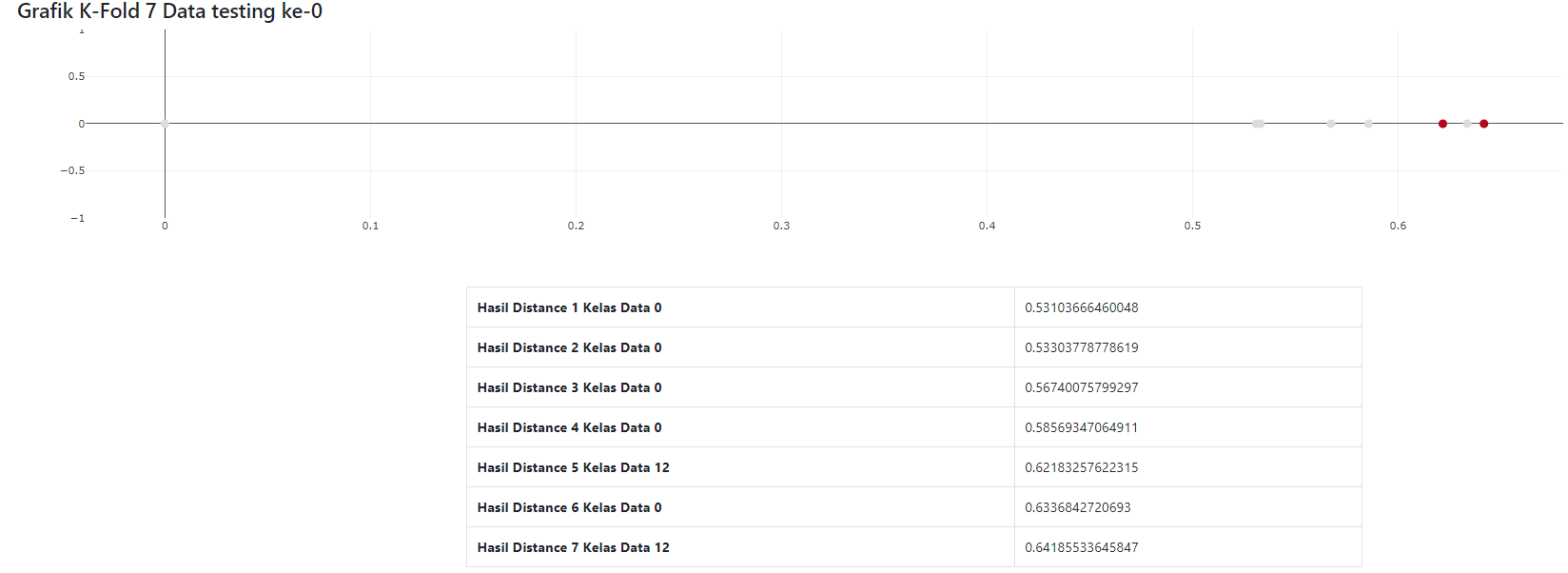


Gambar 4.62. Kelas Data Hasil Klasifikasi yang terpilih salah

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 4.62 menunjukkan kelas label 2 yang sebagai kelas data uji pada data testing ke-6 memilih kelas data terbanyak pertama yang akan dipilih sebagai kelas data hasil klasifikasi dan kelas data hasil klasifikasi KNN saat nilai *k* = 7 di *k–fold* iterasike*–*8 pada data testing ke-6 yang terpilih adalah label 12 yaitu salah.

#### **Identifikasi Pemilihan Kelas pada MKNN saat nilai k = 7 di k–fold iterasi ke–8**

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 4.9. Diperoleh hasil pemilihan kelas yang benar diklasifikasi maupun yang salah diklasifikasi MKNN saat nilai *k* = 7 di *k–fold* iterasike*–*8 sebagai berikut:



Gambar 4.63. Perhitungan *euclidean distance* MKNN sejumlah *k* = 7 pada data testing ke-0

Pada Gambar 4.63 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* = 7 maka dipilih sejumlah 7 nilai *euclidean* *distance*. Diperoleh hasil sebagai berikut:

**Hasil Distance 1 Kelas Data 0 :** 0.53103666460048

**Hasil Distance 2 Kelas Data 0 :** 0.53303778778619

**Hasil Distance 3 Kelas Data 0 :** 0.56740075799297

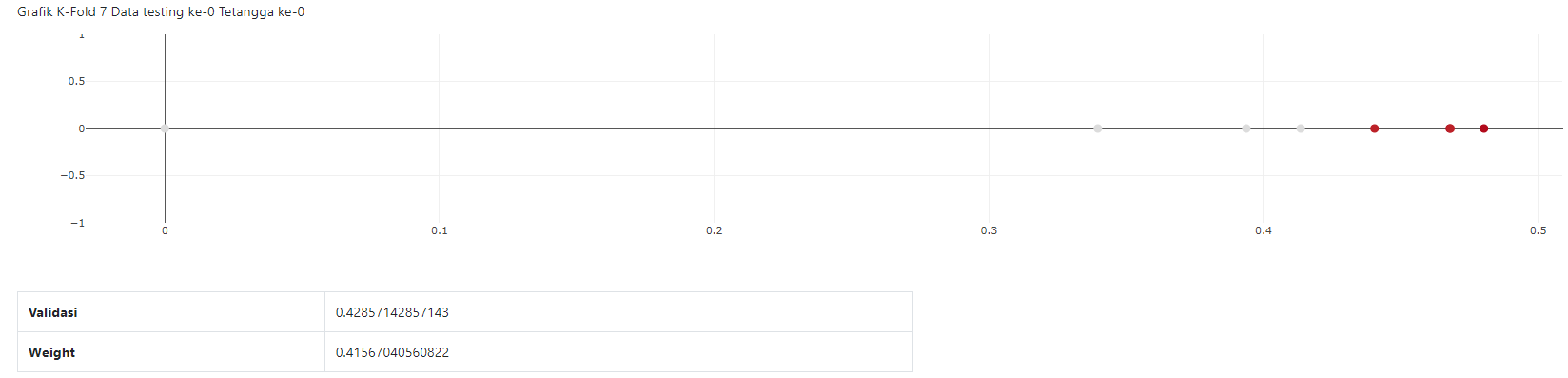
**Hasil Distance 4 Kelas Data 0 :** 0.58569347064911

**Hasil Distance 5 Kelas Data 12 :** 0.62183257622315

**Hasil Distance 6 Kelas Data 0 :** 0.6336842720693

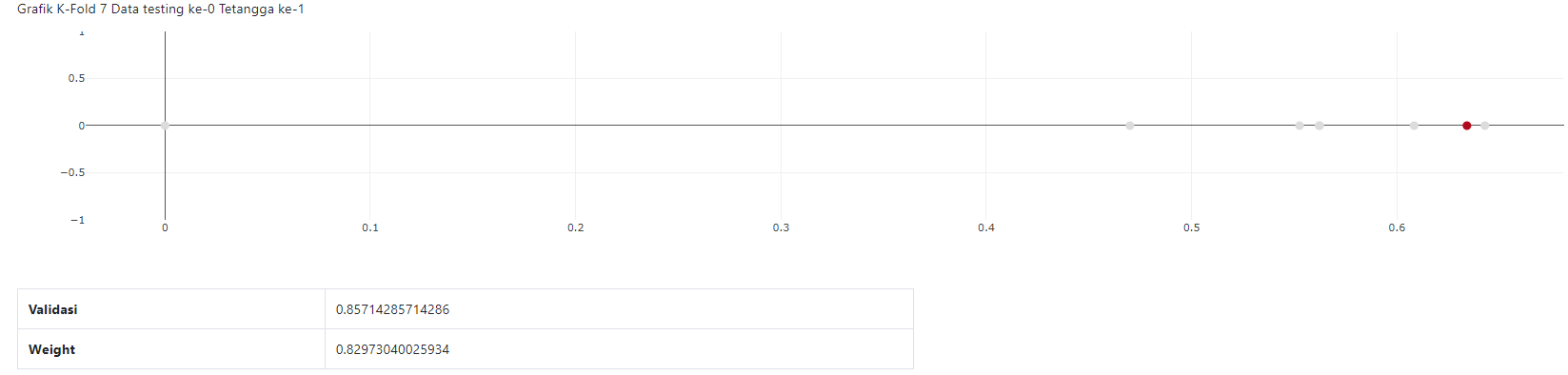
**Hasil Distance 7 Kelas Data 12 :** 0.64185533645847

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* = 7. Jika data latih sama dengan data latih tetangga nya maka bernilai 1, tetapi jika tidak sama dengan data latih tetangga nya 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.



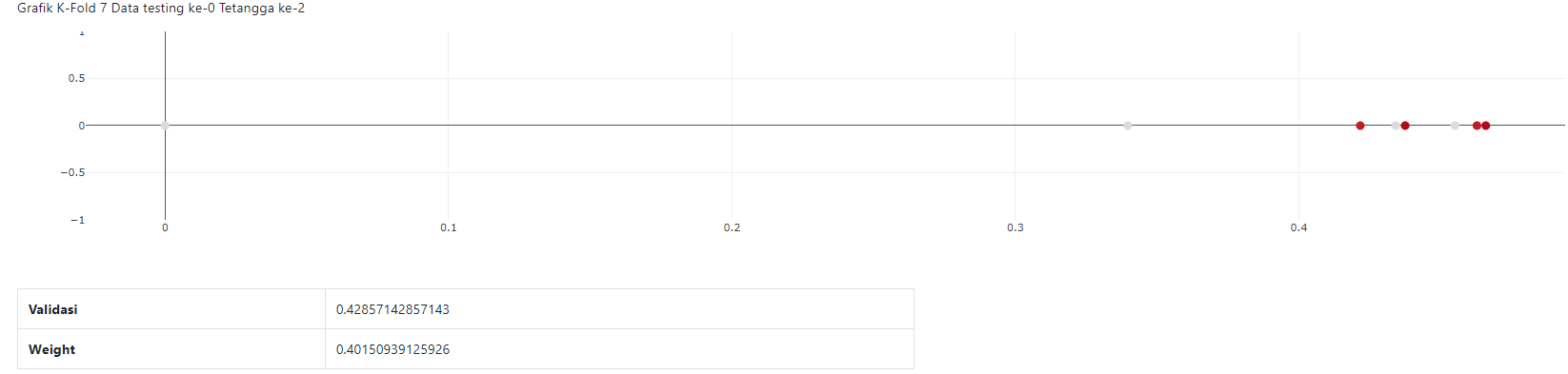
Gambar 4.64. Menghitung validasi dan bobot data testing ke-0 tetangga ke-0

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 4.64 yaitu nilai validasi adalah 0.42857142857143 dan nilai bobot adalah 0.41567040560822.



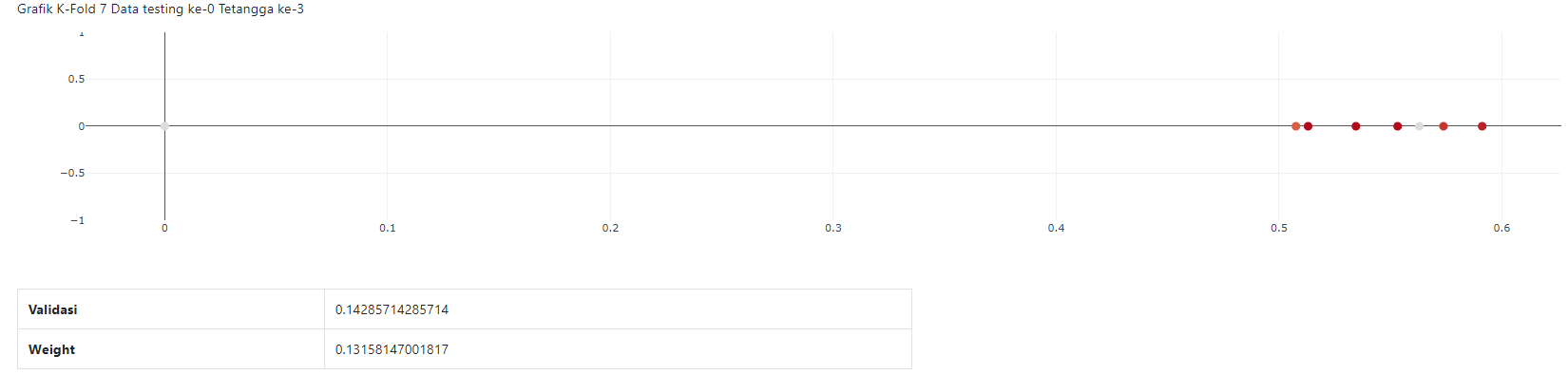
Gambar 4.65. Menghitung validasi dan bobot data testing ke-0 tetangga ke-1

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 4.65 yaitu nilai validasi adalah 0.85714285714286 dan nilai bobot adalah 0.82973040025934.



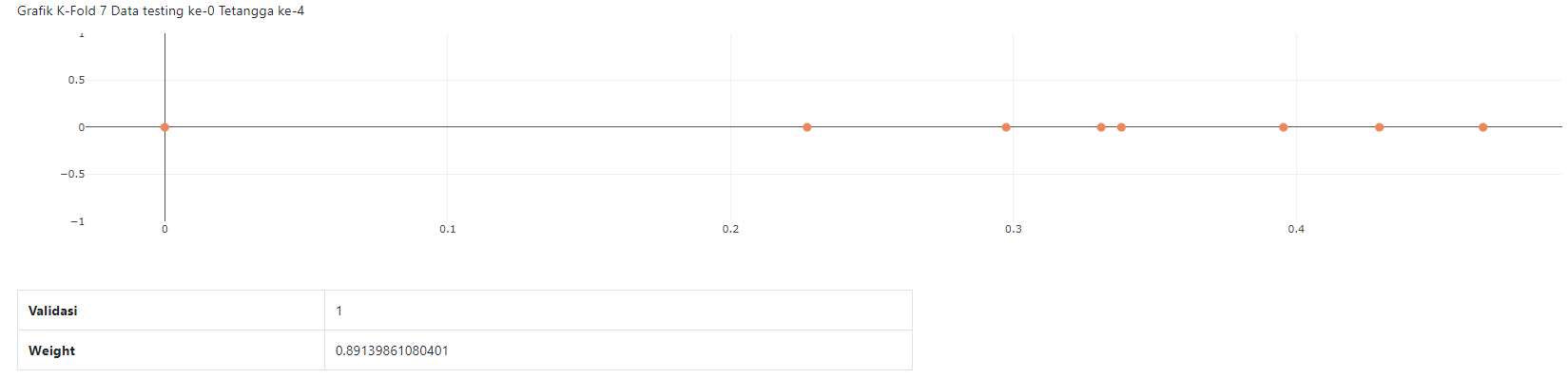
Gambar 4.66. Menghitung validasi dan bobot data testing ke-0 tetangga ke-2

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 4.66 yaitu nilai validasi adalah 0.42857142857143 dan nilai bobot adalah 0.40150939125926.



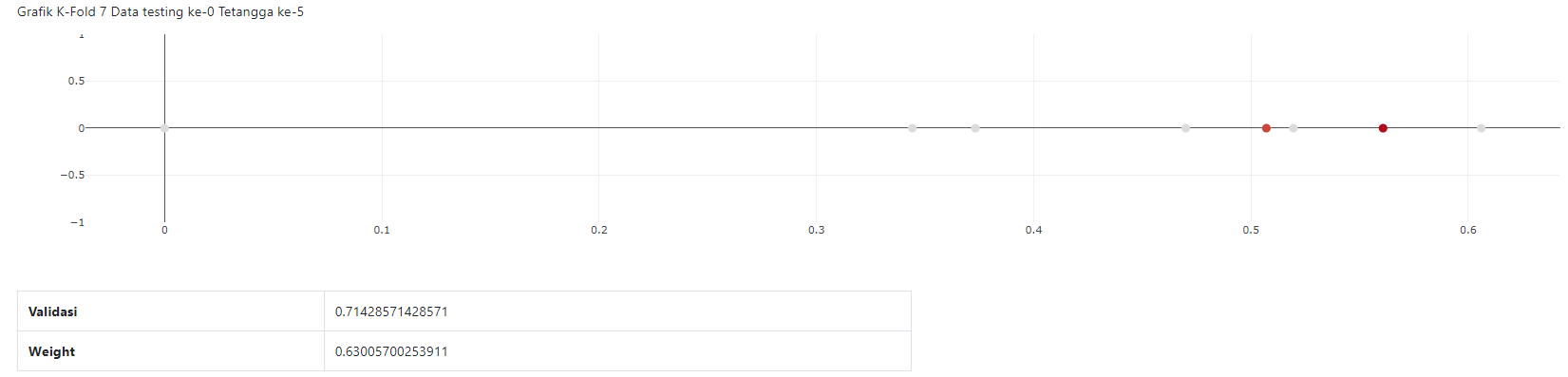
Gambar 4.67. Menghitung validasi dan bobot data testing ke-0 tetangga ke-3

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 4.67 yaitu nilai validasi adalah 0.14285714285714 dan nilai bobot adalah 0.13158147001817.



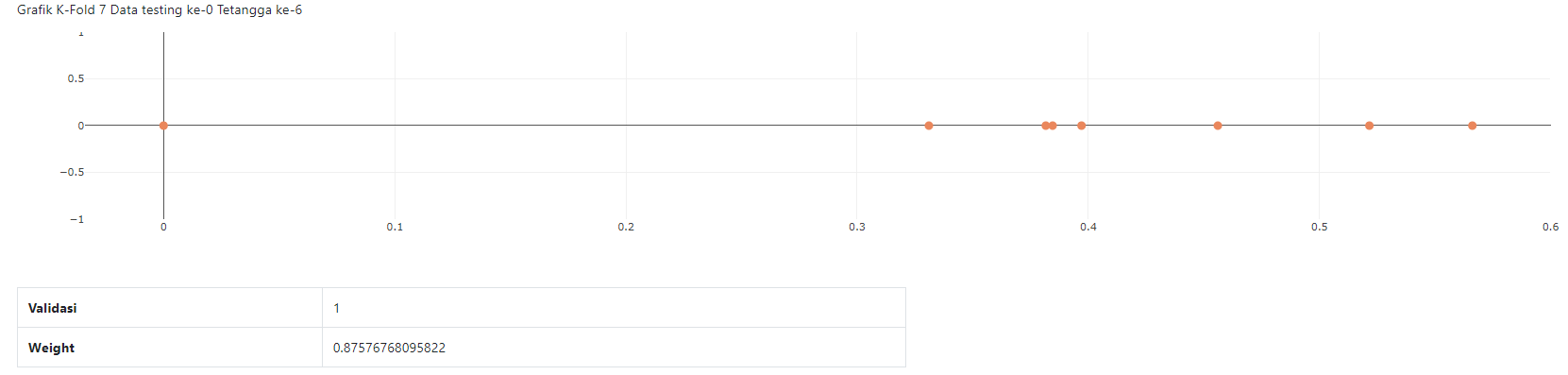
Gambar 4.68. Menghitung validasi dan bobot data testing ke-0 tetangga ke-4

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 4.68 yaitu nilai validasi adalah 1 dan nilai bobot adalah 0.89139861080401.



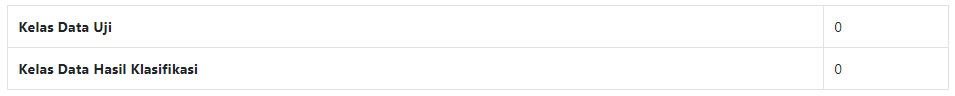
Gambar 4.69. Menghitung validasi dan bobot data testing ke-0 tetangga ke-5

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 4.69 yaitu nilai validasi adalah 0.71428571428571 dan nilai bobot adalah 0.63005700253911.



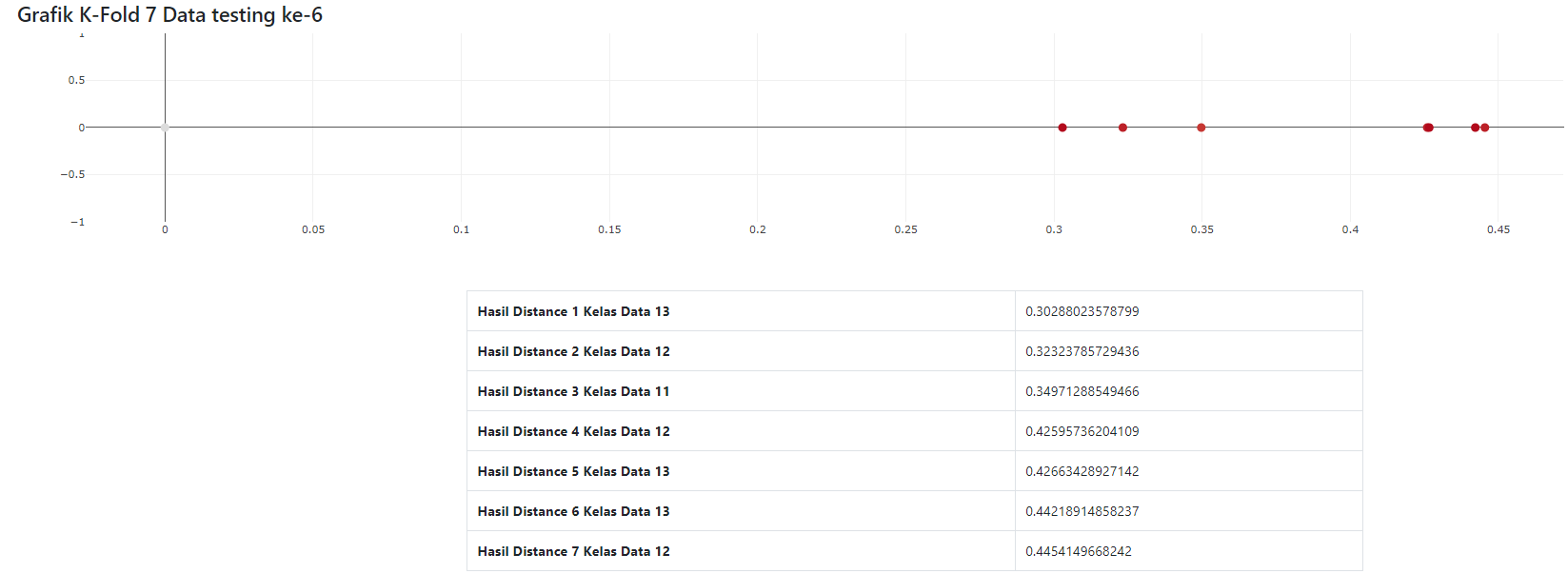
Gambar 4.70. Menghitung validasi dan bobot data testing ke-0 tetangga ke-6

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 4.70 yaitu nilai validasi adalah 1 dan nilai bobot adalah 0.87576768095822.



Gambar 4.71. Kelas Data Hasil Klasifikasi yang terpilih benar

Dari rumus Persamaan (2.12) pada Gambar 4.71 menunjukkan kelas label 0 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* = 7 dan *k–fold* iterasike*–*8 dan data testing ke-0 menghasilkan kelas data hasil klasifikasi yang terpilih adalah label 0 yaitu benar, sedangkan pemilihan label *genre* yang salah pada MKNN sebagai berikut:



Gambar 4.72. Perhitungan *euclidean distance* MKNN sejumlah *k* = 7 pada data testing ke-6

Dilihat pada Gambar 4.72, 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* = 7 hasil *distance* terpilih sejumlah *k* =7 sebagai berikut:

**Hasil Distance 1 Kelas Data 13 :** 0.30288023578799

**Hasil Distance 2 Kelas Data 12 :** 0.32323785729436

**Hasil Distance 3 Kelas Data 11 :** 0.34971288549466

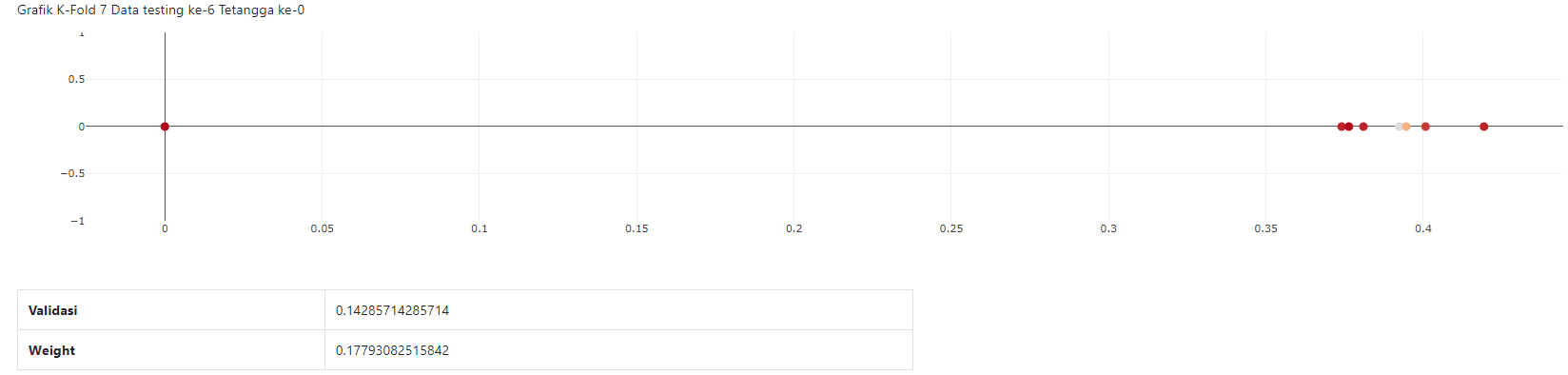
**Hasil Distance 4 Kelas Data 12 :** 0.42595736204109

**Hasil Distance 5 Kelas Data 13 :** 0.42663428927142

**Hasil Distance 6 Kelas Data 13 :** 0.44218914858237

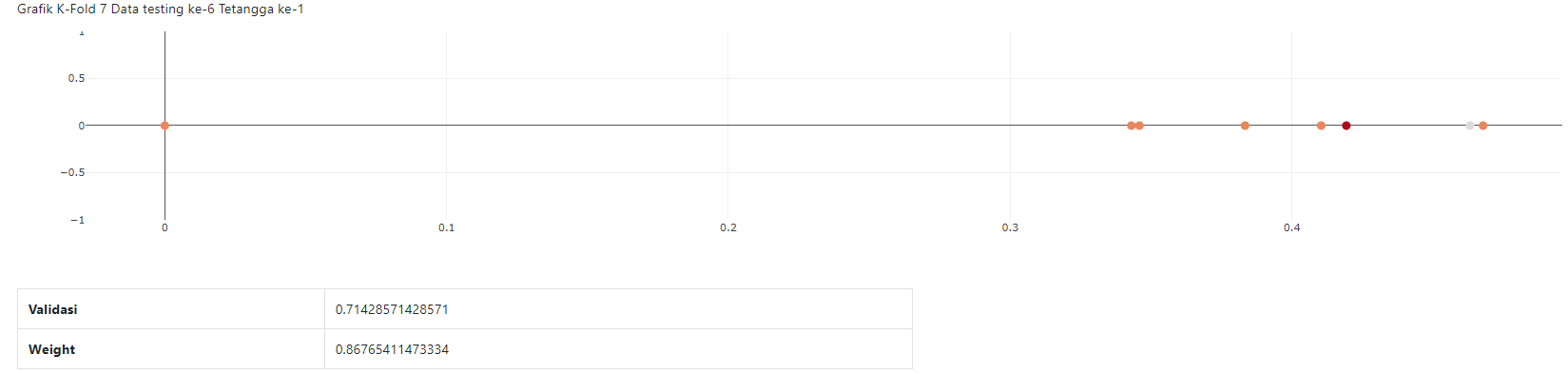
**Hasil Distance 7 Kelas Data 12 :** 0.4454149668242

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* = 7. Jika data latih sama dengan data latih tetangga nya maka bernilai 1, tetapi jika tidak sama dengan data latih tetangga nya 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.



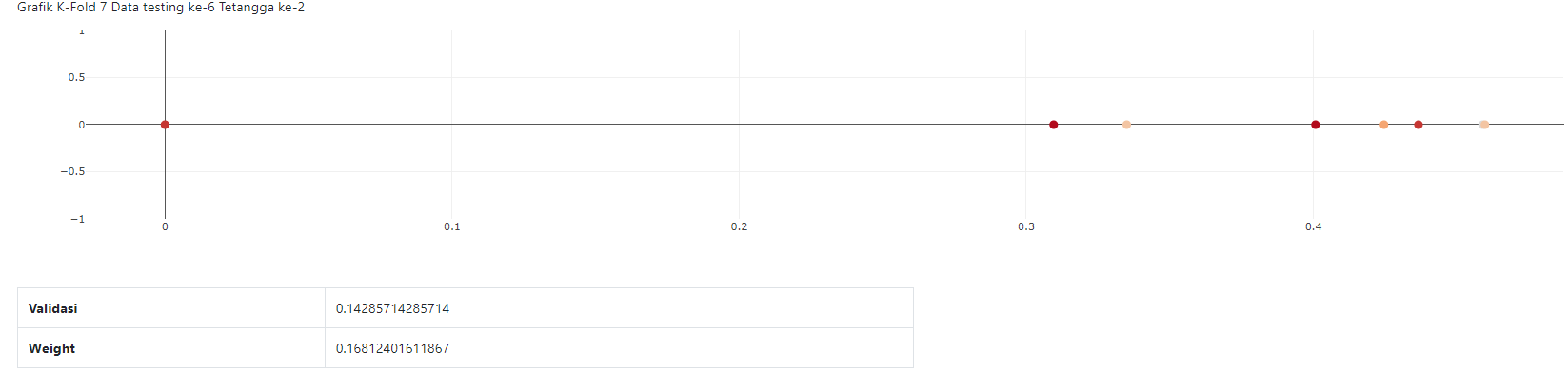
Gambar 4.73. Menghitung validasi dan bobot data testing ke-6 tetangga ke-0

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-0 pada Gambar 4.73 yaitu nilai validasi yang didapatkan adalah 0.14285714285714 dan nilai bobot yang didapatkan adalah 0.17793082515842.



Gambar 4.74. Menghitung validasi dan bobot data testing ke-6 tetangga ke-1

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-1 pada Gambar 4.74 yaitu nilai validasi yang didapatkan adalah 0.71428571428571 dan nilai bobot yang didapatkan adalah 0.86765411473334.



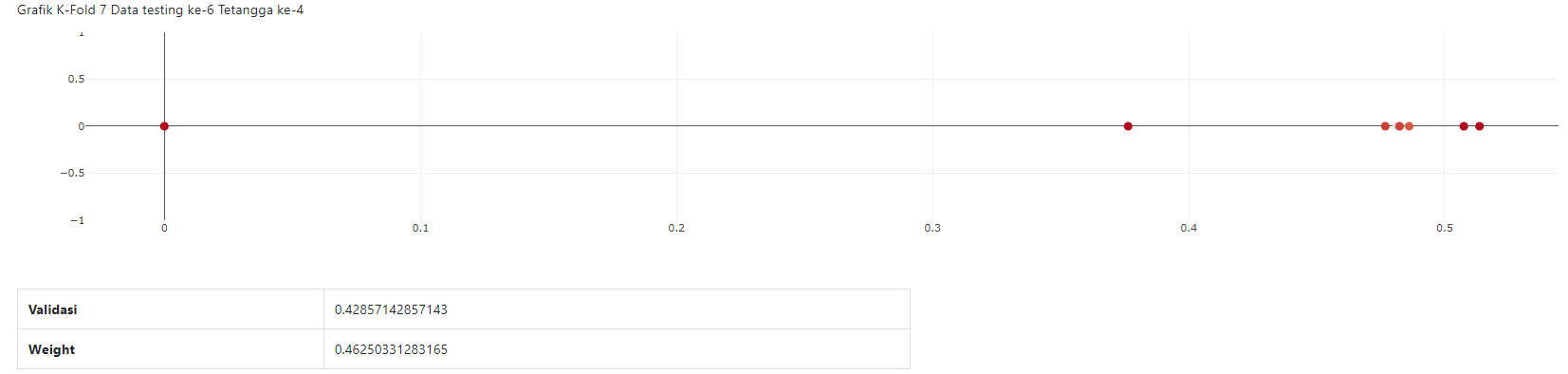
Gambar 4.75. Menghitung validasi dan bobot data testing ke-6 tetangga ke-2

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-2 pada Gambar 4.75 yaitu nilai validasi yang didapatkan adalah 0.14285714285714 dan bobot yang didapatkan adalah 0.16812401611867.



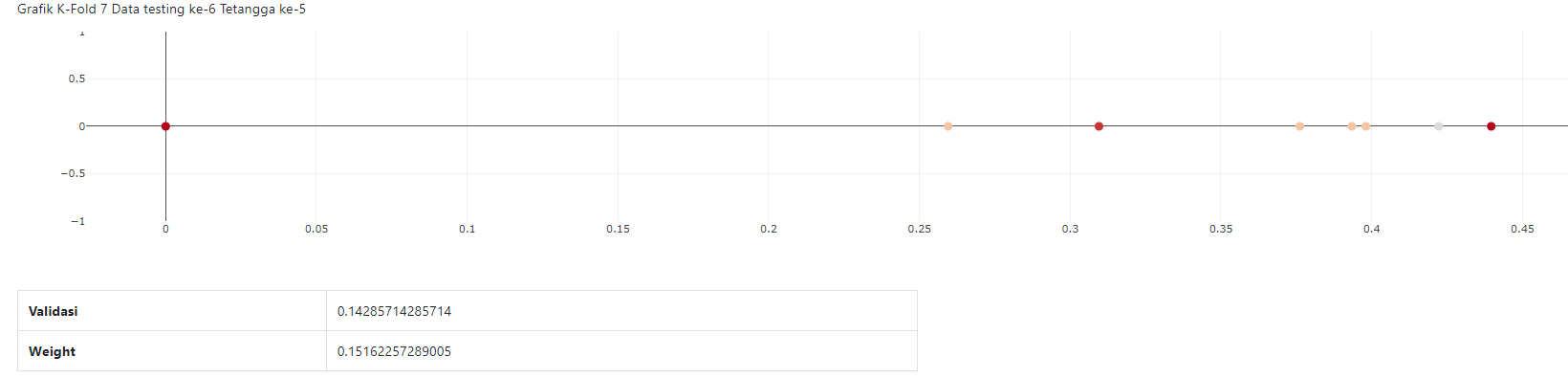
Gambar 4.76. Menghitung validasi dan bobot data testing ke-6 tetangga ke-3

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-3 pada Gambar 4.76 yaitu nilai validasi yang didapatkan adalah 0.57142857142857 dan yang didapatkan adalah 0.61712190523435.



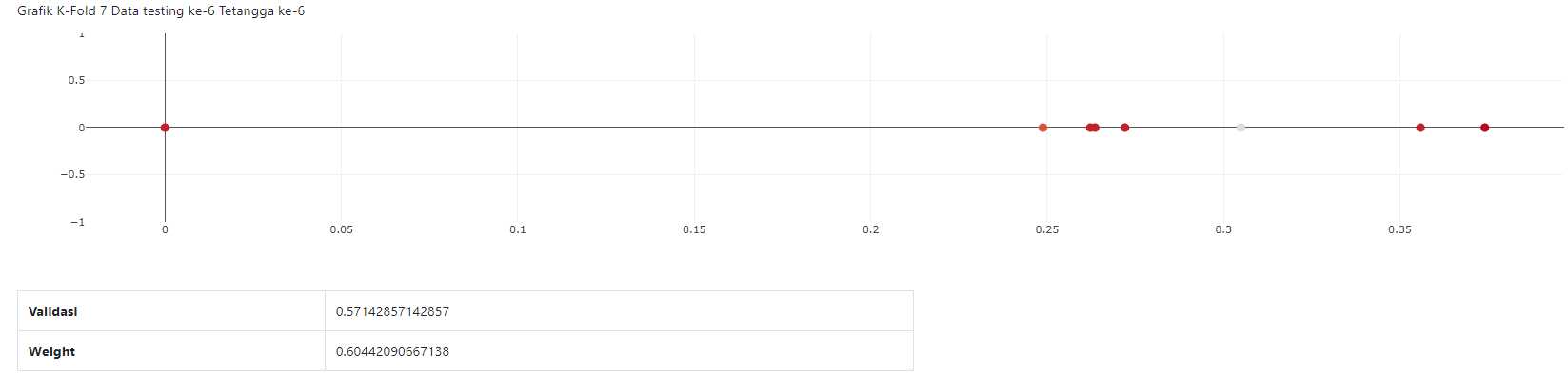
Gambar 4.77. Menghitung validasi dan bobot data testing ke-6 tetangga ke-4

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-4 pada Gambar 4.77 yaitu nilai validasi yang didapatkan adalah 0.42857142857143 dan bobot yang didapatkan adalah 0.46250331283165.



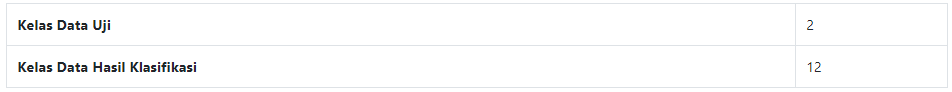
Gambar 4.78. Menghitung validasi dan bobot data testing ke-6 tetangga ke-5

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-3 pada Gambar 4.78 yaitu nilai validasi yang didapatkan adalah 0.14285714285714 dan yang didapatkan adalah 0.15162257289005.



Gambar 4.79. Menghitung validasi dan bobot data testing ke-6 tetangga ke-6

Dari rumus Persamaan (2.9), Persamaan (2.10) dan Persamaan (2.11) diperoleh hasil dari data testing ke-6 dengan tetangga ke-6 pada Gambar 4.79 yaitu nilai validasi yang didapatkan adalah 0.57142857142857 dan bobot yang didapatkan adalah 0.60442090667138.



Gambar 4.80. Kelas Data Hasil Klasifikasi yang terpilih salah

Dari rumus Persamaan (2.12) pada Gambar 4.80 menunjukkan kelas label 2 yang sebagai kelas data uji pada data testing ke-6 memilih bobot yang tertinggi yang akan terpilih sebagai kelas data hasil klasifikasi. Pada klasifikasi MKNN saat nilai *k* = 7 dan *k–fold* iterasike*–*8 dan data testing ke-6 menghasilkan kelas data hasil klasifikasi yang terpilih adalah label 12 yaitu salah.

### **Pengujian Kedua**

Pada pengujian kedua ini, akan dilakukan sama seperti pengujian pertama, tetapi data *genre* pada pengujian pertama akan direduksi menggunakan metode seleksi fitur *Principal Component Analysis* (PCA) menjadi variabel fitur baru dengan cara menghitung nilai *variance* pada Persamaan (2.2), menghitung nilai *covariance* pada Persamaan (2.3), setelah itu menghitung *eigenvalue* dan *eigenvector* pada Persamaan (2.4). Sebelum menghitung *eigenvalue* dan *eigenvector*, nilai dihitung terlebih dahulu pada Persamaan (2.5), selanjutnya mengurutkan *eigenvalue* dan *eigenvector* serta menentukan komponen utama. Kemudian melakukan korelasi antara variabel fitur utama dengan komponen utama pada Persamaan (2.6). Dari Persamaan diatas, hasil reduksi menggunakan PCA pada Lampiran C.1.

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.

#### **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 4.81 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik kembali, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik kembali. Namun nilai rata–rata akurasi metode MKNN tidak melebihi akurasi tertinggi dari PCA*–*KNN.

Pada saat nilai *k* = 16 dan *k* = 17 metode PCA*–*MKNN mendapatkan akurasi tertinggi yang sama dengan nilai akurasi sebesar 25.9%, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 26% pada saat nilai *k* = 16 kemudian nilai akurasi menurun menjadi 25.7% pada saat nilai *k* = 17. Sehingga ketika nilai *k* = 16 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 16 sebanyak 0.1%, sedangkan ketika nilai *k* = 17 (*k* yang paling optimal) metode PCA*–*MKNN memiliki nilai akurasi stabil dari pada metode PCA*–*KNN pada saat nilai *k* = 17.

Selanjutnya pada saat nilai *k* = 29 nilai akurasi naik menjadi 26.3% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 25.6%. 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 0.7%. Pada Gambar 4.81 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA–MKNN dengan PCA*–*KNN pada 1 fitur baru.

Gambar 4.81. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 1 fitur baru

#### **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 rendah dibandingkan dengan PCA*–*KNN. Pada Gambar 4.82 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik kembali, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik kembali. Namun nilai rata–rata akurasi metode PCA*–*MKNN tidak melebihi akurasi tertinggi dari PCA*–*KNN.

Pada saat nilai *k* = 7 dengan nilai akurasi 29.4% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 28%. 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 pada saat nilai *k* = 10 nilai akurasi naik menjadi 31.1% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 29.3%. Sehingga ketika nilai *k* = 10 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 10 sebanyak 1.8%. Pada Gambar 4.82 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 2 fitur baru.

Gambar 4.57. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 2 fitur baru

#### **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 rendah dibandingkan dengan PCA*–*KNN. Pada Gambar 4.83 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik kembali, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik, turun dan naik kembali. Namun nilai rata–rata akurasi metode PCA*–*MKNN tidak melebihi akurasi tertinggi dari PCA*–*KNN.

Pada saat nilai *k* = 7 dengan nilai akurasi 38.4% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 37%. 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 pada saat nilai *k* = 11 nilai akurasi naik menjadi 39.6% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 36.7%. Sehingga ketika nilai *k* = 11 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 11 sebanyak 2.9%. Pada Gambar 4.83 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 3 fitur baru.

Gambar 4.83. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 3 fitur baru

#### **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*–*KNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA*–*MKNN. Pada Gambar 4.84 menunjukkan hasil akurasi PCA*–*MKNN saat *k* bernilai 2 sampai dengan 4 nilai rata–rata akurasi cenderung mengalami kenaikan, selanjutnya hasil akurasi mengalami fluktuasi saat *k* bernilai 5 sampai dengan 29 rata*–*rata akurasi cenderung mengalami tren turun, naik dan turun kembali, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA*–*KNN tidak melebihi akurasi tertinggi dari PCA*–*MKNN.

Pada saat nilai *k* = 4 dengan nilai akurasi 45.1% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 41.6%. Sehingga ketika nilai *k* = 4 (*k* yang paling optimal) metode PCA*–*MKNN meningkatkan nilai akurasi dari metode PCA*–*KNN pada saat nilai *k* = 4 sebanyak 3.5%. Selanjutnya pada saat nilai *k* = 10 nilai akurasi naik menjadi 43.6% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 43.4%. Sehingga ketika nilai *k* = 10 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 10 sebanyak 0.2%. Namun nilai akurasi tersebut tidak melebihi nilai akurasi tertinggi dari metode PCA*–*MKNN. Pada Gambar 4.84 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 4 fitur baru.

Gambar 4.84. Perbandingan Akurasi perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 4 fitur baru

#### **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*–*KNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA*–*MKNN. Pada Gambar 4.85 menunjukkan hasil akurasi PCA*–*MKNN saat *k* bernilai 2 sampai dengan 4 nilai rata–rata akurasi cenderung mengalami kenaikan, selanjutnya hasil akurasi mengalami fluktuasi saat *k* bernilai 5 sampai dengan 29 rata*–*rata akurasi cenderung mengalami tren turun, naik dan turun kembali, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA*–*KNN tidak melebihi akurasi tertinggi dari PCA*–*MKNN.

Pada saat nilai *k* = 4 dengan nilai akurasi 52.1% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 49.6%. Sehingga ketika nilai *k* = 4 (*k* yang paling optimal) metode PCA*–*MKNN meningkatkan nilai akurasi dari metode PCA*–*KNN pada saat nilai *k* = 4 sebanyak 2.5%. Selanjutnya pada saat nilai *k* = 10 nilai akurasi naik menjadi 50.3% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 51.6%. Sehingga ketika nilai *k* = 10 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 10 sebanyak 0.2%. Namun nilai akurasi tersebut tidak melebihi nilai akurasi tertinggi dari metode PCA*–*MKNN. Pada Gambar 4.85 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 5 fitur baru.

Gambar 4.85. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 5 fitur baru

#### **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 4.86 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung 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 saat nilai *k* = 3 dan *k* = 4 metode PCA*–*MKNN mendapatkan akurasi tertinggi yang sama dengan nilai akurasi sebesar 53%, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 51.3% pada saat nilai *k* = 16 kemudian nilai akurasi menurun menjadi 50.6% pada saat nilai *k* = 4. Sehingga ketika nilai *k* = 3 (*k* yang paling optimal) metode PCA*–*MKNN meningkatkan nilai akurasi dari metode PCA*–*KNN pada saat nilai *k* = 3 sebanyak 1.7%, sedangkan ketika nilai *k* = 4 (*k* yang paling optimal) metode PCA*–*MKNN memiliki nilai akurasi stabil dari pada metode PCA*–*KNN pada saat nilai *k* = 17.

Selanjutnya pada saat nilai *k* = 9 nilai akurasi naik menjadi 54.3% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 52%. 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 4.86 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA–MKNN dengan PCA*–*KNN pada 6 fitur baru.

Gambar 4.86. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 6 fitur baru

#### **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 yang lebih rendah dibandingkan dengan PCA*–*KNN. Pada Gambar 4.87 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung 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 saat nilai *k* = 7 dengan nilai akurasi 54.6% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 53%. 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 1.6%. Selanjutnya pada saat nilai *k* = 8 nilai akurasi naik menjadi 55.3% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 53.6%. Sehingga ketika nilai *k* = 8 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 8 sebanyak 1.7%. Pada Gambar 4.87 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 7 fitur baru.

Gambar 4.87. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 7 fitur baru

#### **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 4.88 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung 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 saat nilai *k* = 7 dengan nilai akurasi 54.9% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 55.3%. 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 0.4%. Selanjutnya pada saat nilai *k* = 9 nilai akurasi naik menjadi 56% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 54.1%. 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 1.9%. Pada Gambar 4.88 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 8 fitur baru.

Gambar 4.88. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 8 fitur baru

#### **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*–*KNN memiliki akurasi yang lebih rendah dibandingkan dengan PCA*–*MKNN. Pada Gambar 4.89 menunjukkan hasil akurasi PCA*–*MKNN saat *k* bernilai 2 sampai dengan 4 nilai rata–rata akurasi cenderung mengalami kenaikan, selanjutnya hasil akurasi mengalami fluktuasi saat *k* bernilai 5 sampai dengan 29 rata*–*rata akurasi cenderung mengalami tren turun, naik dan turun kembali, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun. Namun nilai rata–rata akurasi metode PCA*–*KNN tidak melebihi akurasi tertinggi dari PCA*–*MKNN.

Pada saat nilai *k* = 6 metode PCA*–*MKNN mendapatkan akurasi tertinggi dengan nilai akurasi sebesar 58.7%, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 57.9% pada saat nilai *k* = 6 kemudian nilai akurasi naik menjadi 58.4% pada saat nilai *k* = 7, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 57.6% pada saat nilai *k* = 7. Selanjutnya pada saat nilai *k* = 8 nilai akurasi pada metode PCA*–*KNN tidak mengalami perubahan dari akurasi sebelumnya pada saat nilai *k* = 7, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 56.9%.

Sehingga ketika nilai *k* = 6 (*k* yang paling optimal) metode PCA*–*MKNN meningkatkan nilai akurasi dari metode PCA*–*KNN pada saat nilai *k* = 6 sebanyak 0.8%, sedangkan ketika nilai *k* = 7 dan *k* = 8 (*k* yang paling optimal) metode PCA*–*KNN memiliki nilai akurasi stabil dari pada metode PCA*–*KNN pada saat nilai *k* = 7 dan *k* = 8. Pada Gambar 4.89 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA–MKNN dengan PCA*–*KNN pada 9 fitur baru.

Gambar 4.89. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 9 fitur baru

#### **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 4.90 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung 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 saat nilai *k* = 6 dengan nilai akurasi 59.4% merupakan akurasi tertinggi pada metode PCA*–*MKNN, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi sebesar 58.4%. Sehingga ketika nilai *k* = 6 (*k* yang paling optimal) metode PCA*–*MKNN meningkatkan nilai akurasi dari metode PCA*–*KNN pada saat nilai *k* = 6 sebanyak 1.0%. Selanjutnya pada saat nilai *k* = 7 nilai akurasi naik menjadi 60% yang menjadikan akurasi tertinggi pada metode PCA*–*KNN, sedangkan pada metode PCA*–*MKNN nilai akurasi menurun menjadi 58.9%. Sehingga ketika nilai *k* = 7 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 7 sebanyak 1.1%. Pada Gambar 4.90 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 10 fitur baru.

Gambar 4.90. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 10 fitur baru

#### **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 4.91 menunjukkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*KNN cenderung mengalami fluktuasi yang dimana akurasi mengalami tren naik dan turun, sedangkan hasil akurasi saat *k* bernilai 2 sampai dengan 29 nilai rata–rata akurasi pada metode PCA*–*MKNN cenderung 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 saat nilai *k* = 4, *k* = 6 dan *k* = 7 metode PCA*–*MKNN mendapatkan akurasi tertinggi dengan nilai akurasi yang sama sebesar 60.7%, sedangkan pada metode PCA*–*KNN memiliki nilai akurasi yang sama sebesar 61.7% pada saat nilai *k* = 4 dan *k* = 6 kemudian nilai akurasi naik menjadi 63.1% pada saat nilai *k* = 7 yang menjadikan akurasi tertinggi pada metode PCA*–*KNN. Sehingga ketika nilai *k* = 7 (*k* yang paling optimal) metode PCA*–*KNN meningkatkan nilai akurasi dari metode PCA*–*MKNN pada saat nilai *k* = 7 sebanyak 2.4%. Pada Gambar 4.91 disajikan hasil pengujian perbandingan akurasi pada perubahan nilai *k* PCA–MKNN dengan PCA*–*KNN pada 11 fitur baru.

Gambar 4.91. Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNN pada 11 fitur baru

## **Analisis Hasil Pengujian**

Dari subbab 4.1 akan dirangkum hasil pengujian Pengaruh Atribut Lagu yang digunakan terhadap kinerja untuk mengklasifikasi lagu menggunakan KNN dan MKNN ditunjukan pada Tabel 4.10 sebagai berikut.

Tabel 4.10. Pengaruh Atribut Lagu yang digunakan terhadap kinerja untuk mengklasifikasi lagu menggunakan KNN dan MKNN

|  |  |  |
| --- | --- | --- |
| No | Variabel | Pengaruh Atribut Lagu terhadap kinerja untuk mengklasifikasi lagu menggunakan KNN dan MKNN |
| 1 | *Danceability* | Sangat berpengaruh pada penentuan musik *EDM,* *Reggae*, *Soul* dan *Funk* karena lebih cocok untuk menari. |
| 2 | *Valence* | Sangat berpengaruh pada penentuan musik *Classical*, *Country*, *EDM,* *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* karena memiliki emosi yang bervariasi ketimbang musik *Blues* yang cenderung ke emosi negative, sedangkan musik *Funk* memiliki emosi yang gembira. |
| 3 | *Energy* | Sangat berpengaruh pada penentuan musik *Rock* dan *Metal* karena lebih energik. |
| 4 | *Tempo* | Sangat berpengaruh pada penentuan musik *Blues*, *Country*, *Funk*, *Gospel*, *Hip Hop*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul* secara umum hampir sama, namun pada musik *Country*, *Funk* dan *Gospel* memiliki 1 data yang sebaran data *tempo* tinggi di 200 bpm. Selanjutnya untuk musik *Metal* memiliki *tempo* tinggi yang menyebar di 194 bpm sampai 215 bpm, sedangkan musik EDM menyebar rata di 100 bpm sampai 136 bpm. |
| 5 | *Loudness* | Sangat berpengaruh pada penentuan musik *Classical* dan *Jazz* karena lebih lembut dan menenangkan. |
| 6 | *Speechiness* | Sangat berpengaruh pada penentuan musik *Hip Hop* memiliki trek musik dan ucapan vokal seperti musik lagu rap. |
| 7 | *Instrumentalness* | Sangat berpengaruh pada penentuan musik *Classical* dan *Jazz* karena memiliki kecendrungan musik yang tidak disertai vokal. |
| 8 | *Liveness* | Sangat berpengaruh pada penentuan musik *Gospel* karena melakukan rekaman lagu secara *live* (siaran langsung). |
| 9 | *Acousticness* | Sangat berpengaruh pada penentuan musik *Classical*, *Jazz* dan *Pop* karena treknya lebih akustik. |
| 10 | *Key* | Sangat berpengaruh pada penentuan semua *genre* musik karena menggunakan trek kunci tangga nada. |
| 11 | *Mode* | Sangat berpengaruh pada penentuan semua *genre* musik karena menggunakan trek kunci tangga nada minor dan mayor. |
| 12 | *Duration* | Sangat berpengaruh pada penentuan musik *Classical* dan *Funk* karena memiliki durasi lagu yang panjangnya lebih dari 600000 milidetik atau 10 menit. |

Pada pengujian pertama, KNN lebih efektif dibandingkan dengan MKNN pada klasifikasi musik berdasarkan *genre* menggunakan 10*–fold cross–validation*. Hal ini terlihat dari rata–rata akurasi tertinggi dari MKNN yaitu 62% dengan nilai *k* = 5, sedangkan akurasi dari KNN saat nilai *k* = 5 yaitu 63%. Selanjutnya pada saat nilai *k* = 7 akurasi KNN meningkat menjadi 63.3% yang menjadikan ini rata–rata akurasi tertinggi dari KNN, sedangkan saat nilai *k* = 7 pada metode MKNN nilai akurasi cenderung menurun menjadi 61%. Hasil akurasi ini diperoleh dari pengujian rata-rata klasifikasi musik berdasarkan 14 *genre* (*Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul*) yang didapat dengan menggunakan 12 fitur *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode,* dan *duration*.

Sedangkan pada pengujian kedua, KNN juga lebih efektif di sejumlah fitur baru yaitu 1, 2, 3, 6, 7, 8, 10 dan 11 fitur baru dibandingkan dengan MKNN di sejumlah fitur baru yaitu 4, 5 dan 9 fitur baru pada klasifikasi musik berdasarkan *genre* menggunakan metode seleksi fitur *Principal Component Analysis* (PCA). Namun, akurasi yang dihasilkan dari algoritma KNN dan MKNN dengan metode seleksi fitur *Principal Component Analysis* (PCA) belum bisa menghasilkan akurasi yang lebih tinggi daripada akurasi yang tanpa menggunakan PCA dengan 12 fitur.

# **BAB V**

# **KESIMPULAN DAN SARAN**



## **Kesimpulan**

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

1. Pengaruh Atribut Lagu yang digunakan terhadap kinerja untuk mengklasifikasi Lagu menggunakan KNN dan MKNN diantaranya dari data lagu *genre EDM,* *Reggae*, *Soul* dan *Funk* berdasarkan variabelyang digunakantrek nya lebih cocok menari, musik *Funk* memiliki emosi yang bervariasi sementara musik *Blues* cenderung ke emosi negatif, musik rock dan metal lebih energik, musik metal memiliki tempo cepat, musik *Classical* dan *Jazz* lebih lembut dan menenangkan, musik *Hip Hop* trek musik dan ucapan vokal nya seperti musik lagu rap, musik *Classical* dan *Jazz* treknya tidak ada ucapan vokal, musik *Gospel* trek nya lebih live, musik *Classical*, *Jazz* dan *Pop* karena treknya lebih akustik, musik *Classical* dan *Funk* memiliki durasi lagu terlama, dan semua *genre* yang digunakan pada penelitian ini menggunakan trek kunci tangga nada seperti minor dan mayor.
2. Perbandingan performa yang dihasilkan algoritma KNN dan MKNN tanpa menggunakan seleksi fitur *Principal Component Analysis* (PCA) pada klasifikasi 14 *genre* (*Blues*, *Classical*, *Country*, *EDM,* *Funk*, *Gospel*, *Hip Hop*, *Jazz*, *Metal*, *Pop*, *R&B*, *Reggae*, *Rock*, dan *Soul*) dengan menggunakan 12 variabel fitur *danceability, valence, energy, tempo, loudness, speechiness, instrumentalness, liveness, accousticness, key, mode,* dan *duration*. mendapatkan akurasi lebih tinggi.
3. Sedangkan jika menggunakan seleksi fitur *Principal Component Analysis* (PCA), algoritma KNN dan MKNN belum bisa menghasilkan akurasi yang lebih tinggi dari akurasi yang tanpa menggunakan PCA.

## **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 (Anggoro, 2020).
2. Menggunakan metode seleksi fitur lain seperti Shrinkage atau Feature Ablation (Putra, 2020).

# **DAFTAR PUSTAKA**

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

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

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

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

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

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, Ikbal. & 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.

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

Potdar, K. & Kinnerkar, R. (2016). A Comparative Study of Machine Learning Algorithms applied to Predictive Breast Cancer Data. *International Journal of Science and Research (IJSR), Vol 5 Issue 9*, 1550-1553.

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.

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

Subramanian, D. (2019, June 08). *A Simple Introduction to K-Nearest Neighbors Algorithm*. Retrieved from Toward Data Science: https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e

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.

Wibawa, M.S. & Novianti, K.D.V. (2017). Reduksi Fitur Untuk Optimalisasi Klasifikasi Tumor Payudara Berdasarkan Data Citra FNA. *Konferensi Nasional Sistem & Informatika 2017, STMIK STIKOM Bali*, 71-78.

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

Widiantara, W.D. (2018). *Klasifikasi Genre Musik Menggunakan Metode Modified K-Nearest Neighbor.* Riau: Universitas Islam Negeri Sultan Syarif Kasim Riau.

# **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 | 0CcySM3v6xGWPdqkYFPZWZ |
| 3 | 56UsciEGCYrmCTSI3veuvC | 28 | 5tRNgEl5hyFr7E2Adt7dhT |
| 4 | 6IWxHgVbdKyUMydhVGXRaT | 29 | 6bsMl9KBQ7rhFXY0CN9M8o |
| 5 | 31BuUoRtbJwnpzXb4EomGR | 30 | 1SfdzhXyrw84Hs0oMLaeEC |
| 6 | 4dumXvXMddYwmRElxOjxJC | 31 | 1uPb7QAH2vycKlWdGp6ZlY |
| 7 | 0K0lbZb4flVSZUHfV4Uss9 | 32 | 2jXrwWE32uQJBi50xCvE9c |
| 8 | 3v3IWe1q0LwjMR0HlN9V6b | 33 | 3cg0dJfrQB66Qf2YthPb6G |
| 9 | 7BvIryGDGfHGhZBXqDrfrY | 34 | 2Qm98Rw4rdfvo14ZOUiZTN |
| 10 | 0mBSy1Bm4I6sOhe8bjCibK | 35 | 6Gyk7ZHfFWo3d8U7poUEPs |
| 11 | 0gg5d9CX3nwZVy7Hx2Mnuj | 36 | 7cOzgDiU8MLB8IkK1RsWn1 |
| 12 | 0bobD7WpzTRZesNV8C8CER | 37 | 1Msf1vU7lpnbqUDxj6HpnW |
| 13 | 4zBSdZ9E6FWnj7Ks64xdNZ | 38 | 5zGCD8uXutwgpKmXYAVYDb |
| 14 | 1lCQALskBTmdt1HLsqc3oA | 39 | 3OHBGcvkS6oYtCVlxLXP6h |
| 15 | 4QGwPf8fEs72KTSXnPfPHl | 40 | 46tOHo9SThlggrU2MiijEC |
| 16 | 4behsNa8x0dwysWBn6gUHS | 41 | 1Q57HueU3dO9hZB5ALMlPh |
| 17 | 2klvIQb1m0faNZ83PoFB7x | 42 | 2O7OjNCjqflzWOTVg9UlEm |
| 18 | 4FlmEEneXpRCQ0EbFjOYQ5 | 43 | 5cDIbJtehjhImG8eytKwGw |
| 19 | 1i7nQGgwloILcp6ZftH2Qa | 44 | 0utSSFLHA1lfTprfnS7PtS |
| 20 | 3gv5or6v2BIcJMFfb1rHJN | 45 | 4xu0LW34dFEoHG1x65tQPx |
| 21 | 5j6tQBJnNG4wLLdVISJ5g2 | 46 | 1OwuyjjWet6BpgIHSEfnNS |
| 22 | 7ATNFqcr0pTZuFendjxtpk | 47 | 6ihFHZeKzXHIrbY2thI4Vx |
| 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 | 7qNlmpT3ANKIOTiV9J79Vx | 26 | 1DtoFS423q4L3whcGRXJjX |
| 2 | 70mZwr5zctUTtltrA5T2N4 | 27 | 2YjF0o7ytpQ5ECotNJK2ID |
| 3 | 2oLjhx7w8Hyd3gry9cCXr7 | 28 | 3GiSy33Qr6cHArDIJ3Hgrh |
| 4 | 5lM7q5auY0dzqD8jxznujI | 29 | 7Ci53Nn5otFGaccKNRm5qP |
| 5 | 4jep6dMPP0z7HrN375InoL | 30 | 2NPuqw0MFpDe5vccgQesT2 |
| 6 | 23tA7m6wooo2XHJ8YacENk | 31 | 0FwoESTlTuc9leMP6KtYSg |
| 7 | 18tYhf6O5pUMtMnYC8iA6B | 32 | 3a1FmTzjPWRFlh3BE54ff3 |
| 8 | 0S6KWNO0BRK2qs1icIhSHr | 33 | 086sjLPEqdKBgTxbTeCLCv |
| 9 | 0LmBOXeyjnX1zie6AIEQmy | 34 | 1L4NadMj3Jj7YgGaikyhfr |
| 10 | 7AXul8TGnkgrWaMXUNevTq | 35 | 4JYFwr8TFFEiA5Ao5Q7ZS7 |
| 11 | 0l68eSUrRjXY7oO2eKttQ5 | 36 | 1i8tzGQ59LjKqs1DzmAs4z |
| 12 | 0iFgzKBaKbRbG5XJ9ZXLXc | 37 | 3oHSL6pt9LpNrQZuQGu9wL |
| 13 | 3kBtqDJvBkxnmuxo5cQlkj | 38 | 5X4QnmLUFC7AWsxG4tteAm |
| 14 | 0AQJWtbXycfuzKdXLrBuim | 39 | 2SgQu9blQ8O6EBd3Bm8N8L |
| 15 | 22saDmEaFYv0sZsVkZMRNg | 40 | 76ODIwEMw8cuSif210qpb8 |
| 16 | 3h5LXb9kR0qPlpMCO1xfDw | 41 | 0ftAOThcYwpSDapJhDOy7o |
| 17 | 41sr0e0L4r15xTTbU2K0Gx | 42 | 5nioVBwH3xsv10UWdVts10 |
| 18 | 5vvdNx1i8mL32zsqnvptgd | 43 | 4zEGECYEuEPAq52wHswpPB |
| 19 | 0uUH9tgwwfIMXT3XCiQ8NI | 44 | 0uL3g8pq0ZXexhQC0Q4pFA |
| 20 | 6S43QzI1j4znIKuefdpraQ | 45 | 1kxhcIVisIav4uxDoDL8gB |
| 21 | 76CE8P9vX7ev3zjVQmLprs | 46 | 5ejSIiseOusMj76vBVnB5Y |
| 22 | 2e8MxBgVWMSQmxb2zcuCoq | 47 | 6GzMz3s0K1YKwRVI36CgRx |
| 23 | 0uwAWC1AQCXfKEoUnC1mcO | 48 | 2xizRhme7pYeITbH1NLLGt |
| 24 | 1t46GkU2H4x6zG41DRdKd7 | 49 | 294IzNQIRp4soeVaQhows7 |
| 25 | 1ipJdiNMjgvfAilddGT9sI | 50 | 7h6GoPvGHC9uzZJ8bNvfIq |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 0Nu9WA8kEbBWEsay2s8Q0U | 26 | 0ttKO7eBjyoq5Khr2X9lwV |
| 2 | 0gv4JVcgmb8HihoucEXVxh | 27 | 6dGWq08LXuIw6T2oUeHECh |
| 3 | 1OmKo4t4Bh95xQI6WGiUR3 | 28 | 4SBeexFIGHjIo8oK6jUQcw |
| 4 | 4c68Z9wLdHc36y3CNjwQKM | 29 | 0VwTeYNjcl30DyQlt3GPe0 |
| 5 | 3M1aZdmyIqNNFPE8hOKZTZ | 30 | 0MHRECaT0Hzq4iqJutJLtz |
| 6 | 2y849Gnu4VoCnUNaxCh51T | 31 | 7ledv9CTjqdwnvHgRmETyZ |
| 7 | 3rXCZRMiMZp0feGcYXpwYX | 32 | 4TnUKixNWMfajncgdSwFoi |
| 8 | 6sySDQamLYI4wSONik95Rv | 33 | 5Tqb1bof64uhOIRRbqnZkF |
| 9 | 0DUrdV917olRPxTesJAI3n | 34 | 57fGFSzxpGVosCuWWJeKTv |
| 10 | 5cXdB92xagJr4b30GjRbfX | 35 | 5dNGORzxFb5z3v7vSAFkTU |
| 11 | 6ANPIv5r3vdWntrmFa6H6M | 36 | 3OjNkFFZavF89xvRqWCXmU |
| 12 | 1ZGu1039OcTLbDDsFKuY0j | 37 | 63u5QGp9dKXAoYTVKUKBYn |
| 13 | 6JHQBkPuKXCM6j4I2tPqcl | 38 | 6BCrbWBpb8d6KWmEqZ41tr |
| 14 | 0mG66FLhO0iZ6XCAGdjpax | 39 | 4f8hBeMXMvssn6HtFAtblo |
| 15 | 6xatfNMI8NkY5XxRHAeiS4 | 40 | 4afTR1iHGNqIPum5Y9QeOk |
| 16 | 73XG0kGItI3jaHlgxoC5u1 | 41 | 6vmAgl2y9MpoZKrVUXrPe5 |
| 17 | 1OtiKlQ3EALplQWw2LF7Xn | 42 | 4yEvfGgJ9tYfyfXXMLza1V |
| 18 | 7gNXxAvRBZxBhfv0HSUF2z | 43 | 3SRuODYNV4j2ckkJ9Samju |
| 19 | 7qbXjglPXSQW4mNYP3Yk93 | 44 | 2dptTvC3yH12TGtzqp79qw |
| 20 | 1A6P8IxzNyBRsQualaNXNY | 45 | 5ExWY6ojo1Qm3rC7iaEqx2 |
| 21 | 0JSkrv29sBBohFK3qZTEQ0 | 46 | 2MiXC2LfpsoeUP5RvMGEQ3 |
| 22 | 6VXNavmpQoWMoUgI6hyLrc | 47 | 1KNNTdw7SzJ90p6RXq4kGE |
| 23 | 1AEHl2gbtSrdKoLjZX2Y7J | 48 | 71WaAqNtNKNFcHZ2scb9Bi |
| 24 | 5Sqfw0uCMrc2Xam8hXfMMj | 49 | 3wKfQaOgFxZ0qhXjo4iqlI |
| 25 | 3oyByfBYegVEDwYyqx9pRd | 50 | 2jZ0RcOfjdhoyvylgqNrvh |

**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 | 5mt94BslrLXRYh1NPs7M2D | 34 | 1Cq23W4ZxHTY8QbP40qjEc |
| 10 | 4q4HD6OY9aYXEFpPfnfpP4 | 35 | 6JleyZldNFQxcwBL3LgF95 |
| 11 | 3YWf1HtkESNY0Se64USwye | 36 | 7tUTCcqkMCnC9qLqbQeI0T |
| 12 | 0oYWZEjkI86WujE1GfiNZr | 37 | 20McUFi8KCIVdNDLrkTDuZ |
| 13 | 5d7RQFLEiFt1WqOUbEOweq | 38 | 3KcUTtzg9W5Hse7RknCSZr |
| 14 | 5v6aFZhEVli5GOdYG00hQb | 39 | 6dedqDWMV9pSbK3kgF18kv |
| 15 | 2i3lVJNtV5ZNkMo5dHu9AD | 40 | 0eC9r6hu1gAcLkZpSdMi2X |
| 16 | 242WIEMgCkOR1PkAmWDfPI | 41 | 3PKfuv5veJLbC9p4WJvDbG |
| 17 | 5pA6rJ5RAgmXw9QIubTpFk | 42 | 5BgDs2jxeM8cbypzpIGs3z |
| 18 | 5ieQrVW2U70NFMg28mzlqC | 43 | 2gnqqL572vfyt2WPHMjKhO |
| 19 | 3HNs1UAn7OZQiby7DXTIYl | 44 | 46jN5ft1b89T6x8WTq646j |
| 20 | 66W1rVTnEv86dIkFhoiElg | 45 | 2CNOujOA6SenLRhAfzoWKx |
| 21 | 2J6hyfM0x5imlV7lMHwpX2 | 46 | 45ohxHHub9fgz7Oe3Fw0aG |
| 22 | 1HLyQQ3nxbaXwTewU1NCJ1 | 47 | 6mCPTANMqjQBCYm04mTlUp |
| 23 | 0J2vHg5G6Cl7mH5NUK6e1w | 48 | 1LLFoZZ306GC6LZweCtJNQ |
| 24 | 2qzArAFF1cIZJRqWu00ELn | 49 | 3oD19X7tnt8TvRWsqMCp6H |
| 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 | 3ZQZUODe0CoMcgndwZ7cK9 | 33 | 2M2WJ7gBlcKNxdhyfPp9zY |
| 9 | 1r0nn0QF5tNJDtzoIj2mIB | 34 | 2zMJN9JvDlvGP4jB03l1Bz |
| 10 | 5qenhYD4ynkEEHDDwqBiEG | 35 | 13v3siPyvy5TTEZYmGPPse |
| 11 | 4UdJg615OBpJu6MepS6Hq8 | 36 | 0rc9Uwm4vmnQEfx5omLWWF |
| 12 | 2INsiycUgobLSy4ksO4MzK | 37 | 6ztstiyZL6FXzh4aG46ZPD |
| 13 | 3UEh2di2w2D4gfwsvelMn4 | 38 | 2dCmGcEOQrMQhMMS8Vj7Ca |
| 14 | 6EOAOGls9pIdUJsrqytcXK | 39 | 5lA3pwMkBdd24StM90QrNR |
| 15 | 1tFcIERJBVvo4bmtLNSC8W | 40 | 3kim6v8wY7HLPaNmw99SxD |
| 16 | 1TB0dZqFRQsnf4fLLLRKQV | 41 | 5XVb0qfmmkmmsyy7j8oObk |
| 17 | 4o1yx2u3jBmR360WIXZ3L0 | 42 | 4pNiE4LCVV74vfIBaUHm1b |
| 18 | 7jmqTPoxOQsC2w9Cb6z78q | 43 | 4gRA0i5sxx3jAhHaVjPnUN |
| 19 | 7oYHMmBbUF1zDnbgVJwCvC | 44 | 3mRM4NM8iO7UBqrSigCQFH |
| 20 | 6dokyiv4HEy6WWxQCQcOkA | 45 | 6nJh9dyel0o2jmlZzYGh3h |
| 21 | 46eu3SBuFCXWsPT39Yg3tJ | 46 | 28NBmftocOzTPEb6OYA9fW |
| 22 | 1ELKkbWqtKRwvdCBGvETBp | 47 | 0MHXrqn909p0LRTPsNsGEi |
| 23 | 0KDc8l4VNTu6OkMYK9D7HC | 48 | 1qiQduM84A0VeH8Y2uAbqi |
| 24 | 3qQVUOHJdgIFWJd0jrG9GE | 49 | 1i2ZI3tXuJUEjCY6zF3G9w |
| 25 | 3lHLh7um7wGHjBlBVJMA2R | 50 | 7kigmgx2tJJsZHKaa2QC0w |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 1f0ULNih5OLKZk6wcLnoVx | 26 | 6G7VwsjxNBPjCyPmzrUS2R |
| 2 | 6NO0Up4grjlWcprtv4H4dN | 27 | 6v6qYTGVq8zTgHX98SJ7Es |
| 3 | 78jqNs62Tfo3sIxCL2IIpI | 28 | 5hOWTy31TqmFUOGXLcnYQJ |
| 4 | 50rAuUPCO4xCW4cYGY6gRE | 29 | 0FeRIc9264zC1Uo9PdZVf3 |
| 5 | 0GMUJWrXFiasskh8KYbnyw | 30 | 3CC8a83A9FRZ5TUjpONqFq |
| 6 | 4qL8V4pTMJX0HMfyTFWuin | 31 | 05vp57A03E3V6arWWy34yL |
| 7 | 3BLj3BUTTlNjVkhdR45z9e | 32 | 6v9gJ3FeLvhfuInXzSsXpc |
| 8 | 4AgLJxOkacISI93JYPNUtp | 33 | 7wkJJPm4cJnevULnXYJhJm |
| 9 | 4ndpiHFCHD9vMT39QFzkXU | 34 | 04MPnUpNYCSCllb5I2geRn |
| 10 | 35CMJAPksNx7mYVmVKqmio | 35 | 29IGrhmUFETOPkG62HGZsg |
| 11 | 6tzwFtw4AHo2tCexPGMzUN | 36 | 0ekM6xK9H98mJn4c0d3VoC |
| 12 | 5bSlH6lVzXjxnQIIsTPRDn | 37 | 2LgIbzBjl1WCt2Ik5tb1w4 |
| 13 | 3V8cEPvDkFmQh6nUJnG318 | 38 | 4srwFa8t4eIoIMEdGPVFwu |
| 14 | 4wc8FpW9XCUaoxtii3CiCf | 39 | 6vwfHkpyngJeJJxAkA6EoS |
| 15 | 2BiCH1atqXX1OXOtL0PKHS | 40 | 0yay0aA5OwXHYWBww4xuGr |
| 16 | 7hZRgQiZEKf1e8KTEVIkQN | 41 | 4pxjZGz9Wk07smCueT7je3 |
| 17 | 1jVpiofvq33NgRUgx0MpNc | 42 | 0eDajJTpZNVXPUXCqza0HZ |
| 18 | 7thaxfNdPMpWZuqye9AZfX | 43 | 3TI8Bp0MSS9ldP15FdFCqZ |
| 19 | 6rZ18bk8Mipm5T5jmDBqJy | 44 | 1qZ1YyF1uSa4hQIiWlMzH3 |
| 20 | 7HkD5zNeGmDP0ufwr5dgrQ | 45 | 3EBAbFPNcVNDXsgtSnSHDt |
| 21 | 4Smg3uR1jwMr58fmaTTWJC | 46 | 6KvFfXGLtVvW7ZYXJEvvjI |
| 22 | 5AMAF69gvK0v9YT8V6csae | 47 | 2DsTxMYWUOx8NW5hdCEGhm |
| 23 | 3ZLuMo6PrHfOxmD1dLr9c7 | 48 | 6BgWo34t4HQm4cXpNgeTeF |
| 24 | 2RwzuFDm2uqTaMMXXSZSw6 | 49 | 0c4UVGcDiYxFMH0ly0xa4R |
| 25 | 3UnV6ebZwFHLd276t24jbG | 50 | 2lWFN5HviKVkFtybv85FaU |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 78gUO21zFcrzuccs9yEOxL | 26 | 3XNlkb20lYum0okywiW3ry |
| 2 | 5YEOzOojehCqxGQCcQiyR4 | 27 | 3R9j8urSPiBbapNbyuSYkE |
| 3 | 0VgkVdmE4gld66l8iyGjgx | 28 | 49WcLDJOqgpl3yYkRM8AHL |
| 4 | 19gEmPjfqSZT0ulDRfjl0m | 29 | 4l59QWcA9F9qqvNc5pwDtL |
| 5 | 07sOl7WsmCaUqaJaRn9Dss | 30 | 2l8nEO0vsTRCRnGqQty6qx |
| 6 | 4qikXelSRKvoCqFcHLB2H2 | 31 | 2kuZwHNMI6KkMzEcZqrfm1 |
| 7 | 5ujh1I7NZH5agbwf7Hp8Hc | 32 | 4IO8X9W69dIQe0EC5ALXhq |
| 8 | 7t5KciWtlWIS0MXLGqGAVZ | 33 | 5D7LFnpYhd5jlvEWlYLkBS |
| 9 | 5GATP9D34th0D88vlKi38F | 34 | 5yY9lUy8nbvjM1Uyo1Uqoc |
| 10 | 3J87xdiWkf5YYFXXtwOnQO | 35 | 7kRVWQ1G4o8w3ubfNW2ZHB |
| 11 | 3NQhVWYQOF2kHetWDRi1ol | 36 | 3WyRgi8CzQnhzO0xw79tTS |
| 12 | 40pX8i9q4xjOWZR4rGmk2q | 37 | 1jQvUmf16VE9WQCPuK2e8K |
| 13 | 79s5XnCN4TJKTVMSmOx8Ep | 38 | 4TgHt7vKCimpywaiKfl0uj |
| 14 | 5Rgxp3xVqijmMhByj4urIm | 39 | 1qDLTUxg7xyylgtyNleurw |
| 15 | 44ZJ4hLD5onikKsVLQaILr | 40 | 4Y8uVzBLydQgZUPUxxakGJ |
| 16 | 119c93MHjrDLJTApCVGpvx | 41 | 69CQe4ekUZ7txcSfEXYFW9 |
| 17 | 1pyeIeIBIJ8VAJpF0pNPu0 | 42 | 1Si3sqUkvIj6Cz9JYO14OS |
| 18 | 5RGXoZt1qxmAErdhbyKJKq | 43 | 4adrh1YMaPAhtqalhG8hIH |
| 19 | 4IYKjN1DrYzxKXt0umJqsG | 44 | 15Zwn2QjNDVI7xX62tuGVb |
| 20 | 00wjJH0T9jU2l108mHW5Hn | 45 | 3TFO78JhsoNbQUrfKfGdIo |
| 21 | 57hCSu4zMTdnSum7NBL1Ye | 46 | 4Iedi94TIaB2GGb1nMB68v |
| 22 | 6wSwl9xDx28F6bYi7y4JBR | 47 | 3JKgcAa7r07ocVWcV8bS0H |
| 23 | 3WcGdhN6wvGplwVyixNgfI | 48 | 5FVsbdWfHWlLVy0if3lRJ4 |
| 24 | 4iiWcajF1fEUpwcUewc464 | 49 | 0bKz9WAxw6Ct1cHWb2zdFc |
| 25 | 5CcHZZeJAeNW64vnbYGFyM | 50 | 0AVTLkvG3QiVSmIcTW1pVd |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 2ZitZzpm4i01kZm1nNPRPg | 26 | 6V5OcaQior9q7a8r4b93M5 |
| 2 | 0QBgBYJGKbQ94uBKStszv0 | 27 | 3yFlBWuW488dH4Ar8IoV7T |
| 3 | 1e5Xxhz3Ij8TLCNuhB2IEL | 28 | 3oVvVcIpGSDKv2kIHewYsC |
| 4 | 2m6G2vP3x33rQhfmSYB9Iy | 29 | 2o82FM3VkOceMB2UxrYAoa |
| 5 | 1JcaZ6CL3xI6f4sYv9hNbJ | 30 | 7sRYWbFAFNfK59isolP5lc |
| 6 | 0LZ678lgdb9f0ab8AyZBHH | 31 | 55o3rbQcbBVXUqrVszu5rC |
| 7 | 3cAnCLivD2rxdgHN1B27vX | 32 | 2s0lOju4ZXSGnLQeKyRehk |
| 8 | 01Y1rFJpmKXucS9yNdwin0 | 33 | 5pKuAjYVNyUzIgGO6uk0H1 |
| 9 | 4zyETWlER8end95J1QlzXN | 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 | 0ymty2BkJI50ZYOQagxpzt |
| 16 | 09N0r6vIBIEbqcpuAYvKEL | 41 | 5lPhB7GM1jyY1CcnNC2KDo |
| 17 | 0K98aZItWBqCnFiMMiXiD8 | 42 | 1lvIHUFBQoESWGDqTJaxhN |
| 18 | 30qPlOpLeJWD1mASyON1po | 43 | 2KFIDeoTAuwLrXI0ok4jvL |
| 19 | 3JluMpRA2buMEgVjFU8YMj | 44 | 60ToqY3gAOY1eyxX3B13gW |
| 20 | 50ma4LjaP5XDL1BsHrZ1D9 | 45 | 3RpjDd6ITzQAsqG101BHF8 |
| 21 | 7wHsraPGARpUvcRW8holDn | 46 | 3FNtP8FngAuXkPEi4HaFyC |
| 22 | 4Z3UHUWnGLHD0qhf6gSjCO | 47 | 3EeJwALV0aHIW3rIamN6aM |
| 23 | 4GMJNNA04YDBVixAWiK5zD | 48 | 0eo87ddUocYp59TJL6fdks |
| 24 | 1snRw3CQImnuTq7UaX6Are | 49 | 4ENhcl2XYNbXqhiE3eBzon |
| 25 | 4cUBqqGzo3cOpbUnFDcPX2 | 50 | 1Fb9kAuV11nIwb8dVIbvpF |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 5ITT2UsqV9R5jRnpUYebcA | 26 | 5lAXI4GRCeBJYAX821Q329 |
| 2 | 2roO1GmcaLE9iBs0x2D0P0 | 27 | 6qpaNtUKnf9ykKHqHdBhxn |
| 3 | 6QopguYpChUZ4Kp8v4Sm20 | 28 | 2T9hjqYQNAChUAqa9gIRyE |
| 4 | 6tJfZIv8cKoi8dGsaGuZv4 | 29 | 33LekxKniOCWzuSUT4HH6f |
| 5 | 2j4XwCIwdzs0HsMnnB28Sh | 30 | 3Uh8cFbZMYQ*key*PT40K2u3 |
| 6 | 3plQ0DcZjEhWTlu3BPBT9s | 31 | 115WwncSW6yWMcgr93UWXC |
| 7 | 4TFBJMB8TBX3Z3psYnptHL | 32 | 1UmcktVwMddeYqyNfL9WXu |
| 8 | 1MEBTRzovGtlBfHBsOJJC3 | 33 | 2KaDrkD7r8xSgAfHlsxi9U |
| 9 | 0CqCpm4LLQbHUX5ysizWJS | 34 | 7jbEwkIEcmqtAeYEw2k2fB |
| 10 | 54MXF9I8s3DuiQo3g0gZ5k | 35 | 5wq1sRE1xStyFMYibf7lA2 |
| 11 | 6UevPknPIGDy71vgGtZdzP | 36 | 0Dr6tWieJeJ1irEjWteIAv |
| 12 | 6olS0TmHmsGr0hXtcBsiVM | 37 | 0ZWbQZ3tlI7X7XNplTXGX4 |
| 13 | 6jNJY9mDWROqXDJmU6AXnj | 38 | 3Hgn2dn7YMLhsJKlH1BOWa |
| 14 | 4lRvl2PMPqavdm4Q1qNV0M | 39 | 2knnewYZIIaELrwOtXATqc |
| 15 | 0mIUS9GApGWW8wGxNI0CPE | 40 | 34XcuHrlxzt8pTfkZEivwn |
| 16 | 2AjEKcRXkntPkujLXp6hGD | 41 | 23WfrNnWRNPkOqFcwtrjo6 |
| 17 | 3wrUHyoG7VOsCYWHjOnYnl | 42 | 0IvF88MI22klaFnOjuSYmQ |
| 18 | 5bOmleKvMS956pVl9uBLaI | 43 | 1p3733Z0Nv6mg1kMHhSj4W |
| 19 | 18o4pskv7PvFzOWVhShDDh | 44 | 3P1yWZrm1FoDAsEHkBOUN7 |
| 20 | 5Ouuz2mxQIsBMC51fhVAbS | 45 | 7D5nmKDtgj6wP9nWqG4LMk |
| 21 | 7sAAonuVO7tqy1q69X6ZUj | 46 | 5Bog7FbfP8VAqZRBpoic2y |
| 22 | 0iJ8HLpc5I1EifrCho1hPQ | 47 | 0TGtojDTWbq2cTNXJ8R2jM |
| 23 | 178VAE5qT0LSfswN5rTNPu | 48 | 2ok3k2i4xhAbTq45fQ4r65 |
| 24 | 2GKvYk3zl0011ZhbuhdnYv | 49 | 4lsTgoftIh7NR240tfAV82 |
| 25 | 3SqIytsA0RzewQYCJhtIFm | 50 | 2b8OC2ibUQD6ILvEJ6AmBd |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 7puldW6ecGUcM3Vau6U0SD | 26 | 2pV8RpdLemcyMWko8dASVt |
| 2 | 68gbPQJXk2xJQMKsYWKaVc | 27 | 5baXzOMmD0sf26hayRqfqI |
| 3 | 4tzAf07GCR6DlycQkUKlgN | 28 | 3xZMPZQYETEn4hjor3TR1A |
| 4 | 21xRlfE6OjTAFQ4wd9u8IU | 29 | 6yIHGmQLJxWAUZ1ZkENemN |
| 5 | 5IHFSRi1mCCjt4z1Xrht5t | 30 | 7FkH8LndW3WmnwLvVhml4F |
| 6 | 7iqqjQINhAvsNNwXWMU8ga | 31 | 1hVQx6bg4uKPljAUkNjpY2 |
| 7 | 0Ha4WnjeGxmsnzTnlEvjvZ | 32 | 3piY5aMfwEEmXMV6I9kpoi |
| 8 | 3U4isOIWM3VvDubwSI3y7a | 33 | 1D9QqNCGwgKJVVct8fcBpx |
| 9 | 1fq4k7G5kYqUvKIuEmFAAo | 34 | 64GRDrL1efgXclrhVCeuA0 |
| 10 | 3jQyadhLTxpxadQlkFh2b8 | 35 | 58jm1c4IKMFIFoj2Yi3WRB |
| 11 | 78DwRIo6Vj6wqteyerGetr | 36 | 01beCqR9wsVnwzkAJZyTqq |
| 12 | 6CcJMwBtXByIz4zQLzFkKc | 37 | 6Rd4ep779v8CjlFVhaHrNX |
| 13 | 5E5MqaS6eOsbaJibl3YeMZ | 38 | 20IDoqT46mE7o9o1QcYOsV |
| 14 | 0tMzjSDgHIvYf6P97defQP | 39 | 3VK96nMFmP34P4dq2zrAWu |
| 15 | 4fbvXwMTXPWaFyaMWUm9CR | 40 | 2KHllq5yeqjTyXnQPPUPMG |
| 16 | 0u2P5u6lvoDfwTYjAADbn4 | 41 | 3CBxVM0zGj41BQtE6m7gwb |
| 17 | 0pJfsPQesJyCnR5XWZyvj9 | 42 | 671zYm5ZtN7bHaDOwY3Kxy |
| 18 | 3kZC0ZmFWrEHdUCmUqlvgZ | 43 | 6zZMfagNj5Nr8rSdJ9KLqq |
| 19 | 4oeRfmp9XpKWym6YD1WvBP | 44 | 6r2BFEIErNwG0owW4rOQB8 |
| 20 | 2RAH3AVr3KteyBeHixiPQZ | 45 | 1oAWbr6TXkRajJKgSm1vF5 |
| 21 | 3cBtANnJGopPaRMXCl3mV7 | 46 | 0VX8UdmK5BMdExvht2m1Yx |
| 22 | 1C73pwKiXBSJMnSRmbDdB6 | 47 | 0nJW01T7XtvILxQgC5J7Wh |
| 23 | 1QRtDNhyKj9LqLtMhJZPV4 | 48 | 20ebfEtXPLdXv3NNH0D6vo |
| 24 | 0QEmGCfiMQjyDRnRogtMnM | 49 | 7feRZsz7i8C3frJe1I17iR |
| 25 | 07KIEb2h47b9C3ArmzoyML | 50 | 3QmesrvdbPjwf7i40nht1D |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 7N28LOrx4qiEkL1Zd3OTZ3 | 26 | 50DAHcCqneoupFyh96P2GP |
| 2 | 5YLnfy7R2kueN0BRPkjiEG | 27 | 0ZTJrZ1axCUmP1fiFzyabF |
| 3 | 35w6fQ0Wnuv7aIryRx7gTq | 28 | 0cfn5OBGafl32fEsc3z4GE |
| 4 | 2eBD7tEXYvIxrxANueGwtg | 29 | 69mFxANpGyq9s69LqH3d9G |
| 5 | 6JnMvsohvCY0shtvTQCSHd | 30 | 1knipGsDFp6FBFc6aRRyjL |
| 6 | 2LHRwhsMByGIVTtHxlOJg2 | 31 | 6Vas6h9lB5mmeVDXdFqlQA |
| 7 | 1RDLbX9L2G4zDqxXCaMJo5 | 32 | 4jnFqNWeJCeCRHc4HCdxfd |
| 8 | 1FKFLWFa7fVoftc7UmYhDC | 33 | 4LxIGAVfcQIw0zAQRyFhU8 |
| 9 | 1i1ofxObE2LhvDk6mmm3Fo | 34 | 6mz1fBdKATx6qP4oP1I65G |
| 10 | 2K9Ovn1o2bTGqbsABGC6m3 | 35 | 51RvCRzeTqoQgdOBgcpHtT |
| 11 | 0KS2h61pHQ4WmOwruD7uxD | 36 | 4j1Bk0BEIGCF9hR7cSwl9d |
| 12 | 1LV5cAo02H8h5YlZNcjULM | 37 | 5IFCyWplye09HytIP80RCF |
| 13 | 4OmCPaP0sOh2FxuAFdLYzJ | 38 | 7tsRp7QKUBp6hc9bth0h7x |
| 14 | 5XqDJFVCyRTm5J7cIfRmR1 | 39 | 6xkryXuiZU360Lngd4sx13 |
| 15 | 71hBQ9qHsZU2M3vDF8MtaN | 40 | 0JCAezTC5IlK1a9o0YBEsD |
| 16 | 08HPWLi0YmRUStJzkQ1ffh | 41 | 17aVK3oQdgwVG2l0HpP67W |
| 17 | 27PNPVQqaqQ0m2p7uoxnFo | 42 | 5A0aUvnfAEcmUx3hoXjI8R |
| 18 | 7KzLjZBZwm8ikKqswryXWK | 43 | 22RCzyBU4gHnZ6pdJzT3RM |
| 19 | 4cOVTA2GfYTHw99AJDQpHo | 44 | 2eHj0klWkwRQuIrNlPpCPa |
| 20 | 0ri0Han4IRJhzvERHOZTMr | 45 | 6g0Orsxv6glTJCt4cHsRsQ |
| 21 | 3mxXvmaDJXzjZb9WMZVo3L | 46 | 1sZLO7n7ap5F6w9KhCRPPm |
| 22 | 3XG801WWhqC753dekEBkMt | 47 | 2lp8xjq0WTm3HZKHuDEweg |
| 23 | 6QBw7YZIbpo4C9kRlGnsNL | 48 | 7fs4v7plEu5CIsoOt9lLk8 |
| 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 | 1KgsVjdsvVf68nEm8lzvFO |
| 3 | 7HXcSZsNnjm5HV49Rlk7n4 | 28 | 6vADiI54nLUVHXkw2uMMTS |
| 4 | 3Gym3Rtm6FHpTrLlJTcz3j | 29 | 79yFFWjMYYNFq3MHosWlIy |
| 5 | 1JSiX4OT6JUjATNlUA0hL2 | 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 | 4Qup2zholspnhLpgkO77C2 | 40 | 5nDPGDOqmAeSV4hXQsAU01 |
| 16 | 06YsLfaNGjGk0l7Z77H1s9 | 41 | 4FNm4ZJaUVLDCMXcqwS6zY |
| 17 | 0GGPUysb8GpCG76dQH63te | 42 | 30IDv8NTGHirorcjkk8MLR |
| 18 | 7gZAFQROjd8LkSBtAjPiYr | 43 | 5O4erNlJ74PIF6kGol1ZrC |
| 19 | 7BnSsZU3WPrZYj6ZjoX0iE | 44 | 6JRLFiX9NJSoRRKxowlBYr |
| 20 | 3i66eIhRTTULVB6dLJbPCL | 45 | 4tvir8RdYszAMkraiyTSmN |
| 21 | 5lGhbmfAqIbCKIhn5QcIPg | 46 | 49CdYBpfABUa0ZfT8FizQZ |
| 22 | 0di7hMT6o1ogeUN1XDB9Ft | 47 | 5Uv1YNz6uBepPLDdoFANKQ |
| 23 | 044WMdxZWPLXL3hQJmx2We | 48 | 0z0JSkE5Nw3i3nin3BBSuG |
| 24 | 6Q5DtPsZY5I9LEZnjL1PDn | 49 | 5NYWAlsnlOKNUlmKoDMENy |
| 25 | 7578oiL9J6tFHXEjfCXFAS | 50 | 0QkWubamAKEvvcOTriSKX3 |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 5anCkDvJ17aznvK5TED5uo | 26 | 6CDhfV6K4yplWYFR8Ip82v |
| 2 | 3sHH7lklbfpcraDDvYnjo7 | 27 | 1KCAEG6JcsTzi8ddZzs0nk |
| 3 | 6FVYwnVrnAEIRnY3bHJb46 | 28 | 297d7PwfSFHQXCjqqDXQT2 |
| 4 | 0DkmhHO4yyqCJFjosmmWPU | 29 | 0L7zm6afBEtrNKo6C6Gj08 |
| 5 | 7zAt4tdL44D3VuzsvM0N8n | 30 | 5dTHtzHFPyi8TlTtzoz1J9 |
| 6 | 4VymMhILk4DSwghPkLUPVS | 31 | 7eRG7KNDIgjIk3nvykmhOT |
| 7 | 58KPPL1AdLHMvR2O2PZejr | 32 | 1f3yAtsJtY87CTmM8RLnxf |
| 8 | 4wH29TM74O3788Kn1KzWuY | 33 | 4Zc7TCHzuNwL0AFBlyLdyr |
| 9 | 5pKCDm2fw4k6D6C5Rk646C | 34 | 24NXDobrJlwX9ayx7n9OnH |
| 10 | 2zYzyRzz6pRmhPzyfMEC8s | 35 | 003vvx7Niy0yvhvHt4a68B |
| 11 | 4UEo1b0wWrtHMC8bVqPiH8 | 36 | 5moTxUGPZXgGmosl4rIELm |
| 12 | 1fQaoh3imrMunWVZh5kf90 | 37 | 4GJxflCdyvs1q0ciTlLBQJ |
| 13 | 2NImo2MajNai4CLk35wMut | 38 | 7JhXQXjf79vuQoOCtFLDu6 |
| 14 | 5V3mdRI2yQxbSsJGDPc5lD | 39 | 5XnyWvKPVgJsVKmUjFbMv3 |
| 15 | 2SgbR6ttzoNlCRGQOKjrop | 40 | 2dm3UxRHnKk2aN0oZSwULc |
| 16 | 3NW1h8l6EY5N6HNBL83S4V | 41 | 3DM2o4hRRQxdL6WMRpHoPp |
| 17 | 3cwDSDzTiWr5H5xMQhQ6Mx | 42 | 2E7W1X4maFFcjHrVrFA7Vs |
| 18 | 3H0ZY55seypgo0zXI6zwRk | 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 | 01PlEMEsHPNmGEzUGYXo7y |

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

|  |  |  |  |
| --- | --- | --- | --- |
| No | ID | No | ID |
| 1 | 62GYoGszQfROZswLee6W3O | 26 | 66vDBjvjz2YvSF9yJ9JnC5 |
| 2 | 0OKXjd9ZpZioaHyowpQ7as | 27 | 1RG0HipdMCcaW9xJjyKiUb |
| 3 | 6jtiVP3cE6NWVVgEFOkLn7 | 28 | 3SdTKo2uVsxFblQjpScoHy |
| 4 | 6EwrL4VXLw9IV1sG8mIKlA | 29 | 64jxwV2kklovj7ldPsPsXF |
| 5 | 1SBTxJLfZAKvDaz32tUNjn | 30 | 1TEDAg7S7iClCBtpvaQQGA |
| 6 | 6T5UJIY6dG5pYJDHqSfZvG | 31 | 4F2QVxqYaLlsZQx70Wngct |
| 7 | 3ClOMebBxeNTYCOfTH24YJ | 32 | 2j24h8ZvzouJRg2HzHvRH4 |
| 8 | 3M783guFsiqyc4m85YqACf | 33 | 2dGPDINt2l79IbSNyThzyI |
| 9 | 2N5zMZX7YeL1tico8oQxa9 | 34 | 2ADfheRRkK5P2f1t8si3Lj |
| 10 | 0tZkVZ9DeAa0MNK2gY5NtV | 35 | 2OSfEYKhlSsLx6vn4O75RK |
| 11 | 39qYD4J4BKvZMQgxfXl5bv | 36 | 6pJE6B9aAjjAlnS6a5JzfF |
| 12 | 2sXp9Qmvc7mRaDBjBgcGGi | 37 | 0NeJjNlprGfZpeX2LQuN6c |
| 13 | 4qiBDlMzW2vTK8dJKw7xD3 | 38 | 4bEtg70srLrCqQp3Nm2T5R |
| 14 | 1Lc6DaGjgnLxSGLUt2HXzC | 39 | 7ugO7ru8CtlGmX0RIUrQZm |
| 15 | 17ZX2lirfQGxbPLcn25sf7 | 40 | 4591VqUIXysNlmI5NcAIUd |
| 16 | 6t34N78TIbniWvnIwtS2me | 41 | 44UWAmppRuj5KNHWbAASuF |
| 17 | 5rc28C9Ym7zIAypx3QGcnv | 42 | 6IxGVlmZlYIp0RUwKnR6oc |
| 18 | 57jyCG2tQoLpvFMJRIPWLQ | 43 | 2g2GkH3vZHk4lWzBjgQ6nY |
| 19 | 1f08YT5uGGQPYEdJLRz4A8 | 44 | 4eGHlplaq1ME8oetnTuFFf |
| 20 | 3uIE49oWm621AkJVXflKqk | 45 | 4NL8D4RQsC6ux6eI3m3Lg5 |
| 21 | 1E7qCptwRPfvZrIDyDDgjh | 46 | 0BCScVwlR88wcHo5KUwK47 |
| 22 | 6rn7yTWf7iuljTIO047WT4 | 47 | 28T4qj3m5R0qbOYAJzxK6q |
| 23 | 5dXED6MP1v1qghkaniirb1 | 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 | public *function* ViewKlasifikasiKNN($nilaiK){ |
| 2 | $hasilShuffle['nilaiK'] = $nilaiK; |
| 3 | $hasilShuffle['data'] = $this->Data->GetAllShuffle(); |
| 4 | for($f=0; $f<10; $f++){ |
| 5 | $data0 = array(); |
| 6 | $data1 = array(); |
| 7 | $data2 = array(); |
| 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(); |

|  |  |
| --- | --- |
| No | *Source Code* |
| 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, $hasilShuffle['data'][$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, $hasilShuffle['data'][$j]); |
| 62 | array\_push($data1, $hasilShuffle['data'][$j]); |
| 63 | }else{ |
| 64 | $dataIndex1++; |
| 65 | } |
| 66 | } |
| 67 | if($hasilShuffle['data'][$j]->*playlist* == 2){ |
| 68 | if(count($data2)<5&& $dataIndex2 == $f\*5){ |
| 69 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 70 | array\_push($data2, $hasilShuffle['data'][$j]); |
| 71 | }else{ |
| 72 | $dataIndex2++; |
| 73 | } |
| 74 | } |
| 75 | if($hasilShuffle['data'][$j]->*playlist* == 3){ |
| 76 | if(count($data3)<5&& $dataIndex3 == $f\*5){ |
| 77 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 78 | array\_push($data3, $hasilShuffle['data'][$j]); |
| 79 | }else{ |
| 80 | $dataIndex3++; |
| 81 | } |
| 82 | } |
| 83 | if($hasilShuffle['data'][$j]->*playlist* == 4){ |
| 84 | if(count($data4)<5&& $dataIndex4 == $f\*5){ |
| 85 | array\_push($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){ |

|  |  |
| --- | --- |
| 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, $hasilShuffle['data'][$j]); |
| 126 | array\_push($data9, $hasilShuffle['data'][$j]); |
| 127 | }else{ |
| 128 | $dataIndex9++; |
| 129 | } |
| 130 | } |
| 131 | if($hasilShuffle['data'][$j]->*playlist* == 10){ |
| 132 | if(count($data10)<5&& $dataIndex10 == $f\*5){ |
| 133 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 134 | array\_push($data10, $hasilShuffle['data'][$j]); |
| 135 | }else{ |
| 136 | $dataIndex10++; |
| 137 | } |
| 138 | } |
| 139 | if($hasilShuffle['data'][$j]->*playlist* == 11){ |
| 140 | if(count($data11)<5&& $dataIndex11 == $f\*5){ |
| 141 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 142 | array\_push($data11, $hasilShuffle['data'][$j]); |
| 143 | }else{ |
| 144 | $dataIndex11++; |
| 145 | } |
| 146 | } |
| 147 | if($hasilShuffle['data'][$j]->*playlist* == 12){ |
| 148 | if(count($data12)<5&& $dataIndex12 == $f\*5){ |
| 149 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 150 | array\_push($data12, $hasilShuffle['data'][$j]); |
| 151 | }else{ |
| 152 | $dataIndex12++; |
| 153 | } |
| 154 | } |
| 155 | if($hasilShuffle['data'][$j]->*playlist* == 13){ |
| 156 | if(count($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); |
| 166 | $jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 167 | for($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 | pow(($dataTesting[$j]->instrumentalness) - ($hasilShuffle['data'][$k]->instrumentalness), 2)+ |
| 190 | pow(($dataTesting[$j]->liveness) - ($hasilShuffle['data'][$k]->liveness), 2)+ |
| 191 | pow(($dataTesting[$j]->valence) - ($hasilShuffle['data'][$k]->valence), 2)+ |
| 192 | pow(($dataTesting[$j]->tempo) - ($hasilShuffle['data'][$k]->tempo), 2)+ |
| 193 | pow(($dataTesting[$j]->duration) - ($hasilShuffle['data'][$k]->duration), 2)); |
| 194 | array\_push($hasilDistance, $kalkulasiDistance); |
| 195 | } |
| 196 | } |
| 197 | usort($hasilDistance, "cmp"); |
| 198 | for($k=0; $k<$nilaiK; $k++){ |
| 199 | $hasilShuffle['hasilDistance'][$f][$j][$k] = $hasilDistance[$k]; |
| 200 | } |
| 201 | $hasilClass = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 202 | for($k=0;$k<$nilaiK;$k++){ |
| 203 | $hasilClass[$hasilShuffle['data'][$hasilDistance[$k]->no-1]->playlist] +=1; |
| 204 | } |
| 205 | $max = $hasilClass[0]; |
| 206 | $classHigh = 0; |
| 207 | for($k=1;$k<count($hasilClass);$k++){ |
| 208 | if($max<$hasilClass[$k]){ |
| 209 | $max = $hasilClass[$k]; |
| 210 | $classHigh = $k; |
| 211 | } |
| 212 | } |
| 213 | $hasilShuffle['dataHasilTesting'][$f][$j] = $classHigh; |
| 214 | if($classHigh == $hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist){ |
| 215 | $jumlahBenar[$hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist]++; |
| 216 | }else { |
| 217 | $jumlahSalah[$hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist]++; |
| 218 | } |
| 219 | switch ($hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist) { |
| 220 | case 0: |
| 221 | $dataHasil0[$classHigh]++; |
| 222 | break; |
| 223 | case 1: |

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

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

**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 | *Source Code* |
| 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(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 37 | $hasilData1 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 38 | $hasilData2 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 39 | $hasilData3 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 40 | $hasilData4 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 41 | $hasilData5 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 42 | $hasilData6 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 43 | $hasilData7 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 44 | $hasilData8 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 45 | $hasilData9 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 46 | $hasilData10 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 47 | $hasilData11 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |

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

|  |  |
| --- | --- |
| No | *Source Code* |
| 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 == $g\*5){ |
| 93 | array\_push($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 == $g\*5){ |
| 101 | array\_push($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 == $g\*5){ |
| 109 | array\_push($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 == $g\*5){ |
| 117 | array\_push($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 == $g\*5){ |
| 125 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 126 | array\_push($data9, $hasilShuffle['data'][$j]); |
| 127 | }else{ |
| 128 | $dataIndex9++; |
| 129 | } |
| 130 | } |
| 131 | if($hasilShuffle['data'][$j]->*playlist* == 10){ |
| 132 | if(count($data10)<5&& $dataIndex10 == $g\*5){ |
| 133 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 134 | array\_push($data10, $hasilShuffle['data'][$j]); |
| 135 | }else{ |
| 136 | $dataIndex10++; |
| 137 | } |
| 138 | } |
| 139 | if($hasilShuffle['data'][$j]->*playlist* == 11){ |
| 140 | if(count($data11)<5&& $dataIndex11 == $g\*5){ |
| 141 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 142 | array\_push($data11, $hasilShuffle['data'][$j]); |
| 143 | }else{ |
| 144 | $dataIndex11++; |
| 145 | } |
| 146 | } |
| 147 | if($hasilShuffle['data'][$j]->*playlist* == 12){ |
| 148 | if(count($data12)<5&& $dataIndex12 == $g\*5){ |
| 149 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 150 | array\_push($data12, $hasilShuffle['data'][$j]); |
| 151 | }else{ |
| 152 | $dataIndex12++; |
| 153 | } |
| 154 | } |
| 155 | if($hasilShuffle['data'][$j]->*playlist* == 13){ |
| 156 | if(count($data13)<5&& $dataIndex13 == $g\*5){ |

|  |  |
| --- | --- |
| No | *Source Code* |
| 157 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 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); |
| 166 | $jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 167 | for($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 | pow(($dataTesting[$j]->instrumentalness) - ($hasilShuffle['data'][$k]->instrumentalness), 2)+ |
| 190 | pow(($dataTesting[$j]->liveness) - ($hasilShuffle['data'][$k]->liveness), 2)+ |
| 191 | pow(($dataTesting[$j]->valence) - ($hasilShuffle['data'][$k]->valence), 2)+ |
| 192 | pow(($dataTesting[$j]->tempo) - ($hasilShuffle['data'][$k]->tempo), 2)+ |
| 193 | pow(($dataTesting[$j]->duration) - ($hasilShuffle['data'][$k]->duration), 2)); |
| 194 | array\_push($hasilDistance, $kalkulasiDistance); |
| 195 | } |
| 196 | } |
| 197 | usort($hasilDistance, "cmp"); |
| 198 | for($k=0; $k<$nilaiK; $k++){ |
| 199 | $hasilShuffle['hasilDistance'][$g][$j][$k] = $hasilDistance[$k]; |
| 200 | } |
| 201 | $weightResult = array(); |
| 202 | for($i=0;$i<$nilaiK;$i++){ |
| 203 | $dataTest = $hasilShuffle['data'][$hasilDistance[$i]->no-1]; |
| 204 | $hasilShuffle['dataTestingResult'][$g][$j][$i] = $dataTest; |
| 205 | $resultDistance = array(); |
| 206 | for($k=0;$k<count($hasilShuffle['data']);$k++){ |
| 207 | $sama = 0; |
| 208 | for($l=0; $l<count($allDataTesting);$l++){ |
| 209 | if($allDataTesting[$l]->no == $hasilShuffle['data'][$k]->no){ |
| 210 | $sama = 1; |
| 211 | break; |
| 212 | } |
| 213 | } |
| 214 | if($sama == 0 && $dataTest->no != $hasilShuffle['data'][$k]->no){ |
| 215 | $kalDistance = new MyDistance(); |
| 216 | $kalDistance->no = $hasilShuffle['data'][$k]->no; |
| 217 | $kalDistance->distance = sqrt( |
| 218 | pow(($dataTest->danceability) - ($hasilShuffle['data'][$k]->danceability), 2)+ |
| 219 | pow(($dataTest->energy) - ($hasilShuffle['data'][$k]->energy), 2)+ |
| 220 | pow(($dataTest->key) - ($hasilShuffle['data'][$k]->key), 2)+ |
| 221 | pow(($dataTest->loudness) - ($hasilShuffle['data'][$k]->loudness), 2)+ |

|  |  |
| --- | --- |
| No | *Source Code* |
| 222 | pow(($dataTest->mode) - ($hasilShuffle['data'][$k]->mode), 2)+ |
| 223 | pow(($dataTest->speechiness) - ($hasilShuffle['data'][$k]->speechiness), 2)+ |
| 224 | pow(($dataTest->accousticness) - ($hasilShuffle['data'][$k]->accousticness), 2)+ |
| 225 | pow(($dataTest->instrumentalness) - ($hasilShuffle['data'][$k]->instrumentalness), 2)+ |
| 226 | pow(($dataTest->liveness) - ($hasilShuffle['data'][$k]->liveness), 2)+ |
| 227 | pow(($dataTest->valence) - ($hasilShuffle['data'][$k]->valence), 2)+ |
| 228 | pow(($dataTest->tempo) - ($hasilShuffle['data'][$k]->tempo), 2)+ |
| 229 | pow(($dataTest->duration) - ($hasilShuffle['data'][$k]->duration), 2)); |
| 230 | array\_push($resultDistance, $kalDistance); |
| 231 | } |
| 232 | } |
| 233 | usort($resultDistance, "cmp"); |
| 234 | for($k=0; $k<$nilaiK; $k++){ |
| 235 | $hasilShuffle['resultDistance'][$g][$j][$i][$k] = $resultDistance[$k]; |
| 236 | } |
| 237 | $validasi = 0; |
| 238 | for($k=0; $k<$nilaiK; $k++){ |
| 239 | if($hasilShuffle['data'][$resultDistance[$k]->no-1]->playlist == $dataTest->playlist){ |
| 240 | $validasi++; |
| 241 | } |
| 242 | } |
| 243 | $validasi = $validasi/$nilaiK; |
| 244 | $hasilShuffle['Validasi'][$g][$j][$i] = $validasi; |
| 245 | $cacheWeight = $validasi \* (1/($hasilDistance[$i]->distance+0.5)); |
| 246 | $hasilShuffle['Weight'][$g][$j][$i] = $cacheWeight; |
| 247 | $classWight = new MyWeight(); |
| 248 | $classWight->playlist = $dataTest->playlist; |
| 249 | $classWight->weight = $cacheWeight; |
| 250 | array\_push($weightResult, $classWight); |
| 251 | } |
| 252 | $hasilClassDistance = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 253 | for($k=0;$k<count($weightResult);$k++){ |

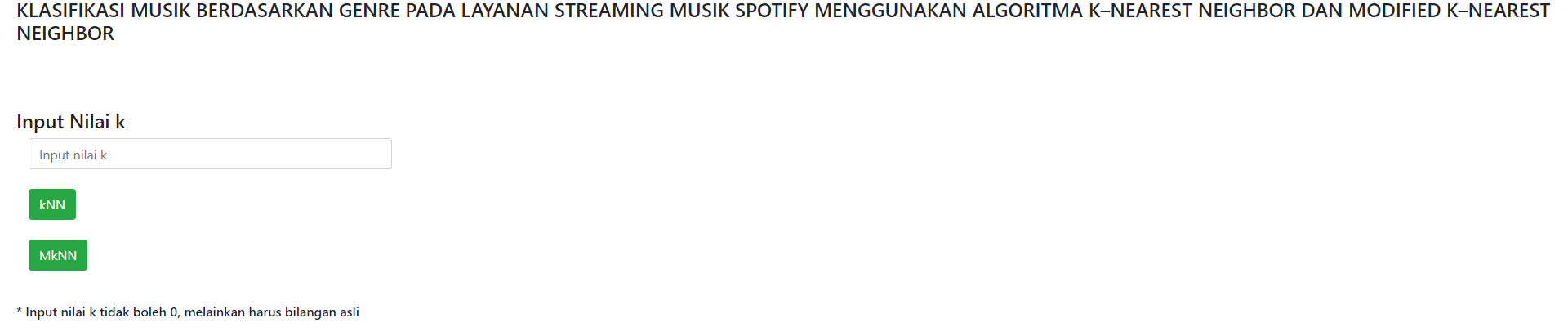
|  |  |
| --- | --- |
| No | *Source Code* |
| 254 | $hasilClassDistance[$weightResult[$k]->*playlist*]+=$weightResult[$k]->weight; |
| 255 | } |
| 256 | $choseClass = 0; |
| 257 | $cacheMax = 0; |
| 258 | for($k=0;$k<count($hasilClassDistance);$k++){ |
| 259 | if($cacheMax<$hasilClassDistance[$k]){ |
| 260 | $cacheMax = $hasilClassDistance[$k]; |
| 261 | $choseClass = $k; |
| 262 | } |
| 263 | } |
| 264 | $hasilShuffle['dataHasilTesting'][$g][$j] = $choseClass; |
| 265 | if($choseClass == $hasilShuffle['data'][$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 | $hasilData0[$choseClass]++; |
| 273 | break; |
| 274 | case 1: |
| 275 | $hasilData1[$choseClass]++; |
| 276 | break; |
| 277 | case 2: |
| 278 | $hasilData2[$choseClass]++; |
| 279 | break; |
| 280 | case 3: |
| 281 | $hasilData3[$choseClass]++; |
| 282 | break; |
| 283 | case 4: |
| 284 | $hasilData4[$choseClass]++; |
| 285 | break; |
| 286 | case 5: |
| 287 | $hasilData5[$choseClass]++; |
| 288 | break; |
| 289 | case 6: |
| 290 | $hasilData6[$choseClass]++; |

|  |  |
| --- | --- |
| No | *Source Code* |
| 291 | break; |
| 292 | case 7: |
| 293 | $hasilData7[$choseClass]++; |
| 294 | break; |
| 295 | case 8: |
| 296 | $hasilData8[$choseClass]++; |
| 297 | break; |
| 298 | case 9: |
| 299 | $hasilData9[$choseClass]++; |
| 300 | break; |
| 301 | case 10: |
| 302 | $hasilData10[$choseClass]++; |
| 303 | break; |
| 304 | case 11: |
| 305 | $hasilData11[$choseClass]++; |
| 306 | break; |
| 307 | case 12: |
| 308 | $hasilData12[$choseClass]++; |
| 309 | break; |
| 310 | case 13: |
| 311 | $hasilData13[$choseClass]++; |
| 312 | break; |
| 313 | } |
| 314 | } |
| 315 | $hasilShuffle["DataMatriks"][$g][0] = $hasilData0; |
| 316 | $hasilShuffle["DataMatriks"][$g][1] = $hasilData1; |
| 317 | $hasilShuffle["DataMatriks"][$g][2] = $hasilData2; |
| 318 | $hasilShuffle["DataMatriks"][$g][3] = $hasilData3; |
| 319 | $hasilShuffle["DataMatriks"][$g][4] = $hasilData4; |
| 320 | $hasilShuffle["DataMatriks"][$g][5] = $hasilData5; |
| 321 | $hasilShuffle["DataMatriks"][$g][6] = $hasilData6; |
| 322 | $hasilShuffle["DataMatriks"][$g][7] = $hasilData7; |
| 323 | $hasilShuffle["DataMatriks"][$g][8] = $hasilData8; |
| 324 | $hasilShuffle["DataMatriks"][$g][9] = $hasilData9; |
| 325 | $hasilShuffle["DataMatriks"][$g][10] = $hasilData10; |
| 326 | $hasilShuffle["DataMatriks"][$g][11] = $hasilData11; |
| 327 | $hasilShuffle["DataMatriks"][$g][12] = $hasilData12; |
| 328 | $hasilShuffle["DataMatriks"][$g][13] = $hasilData13; |
| 329 | $calculateBenar = 0; |
| 330 | for($j=0; $j<count($jumlahBenar); $j++){ |

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

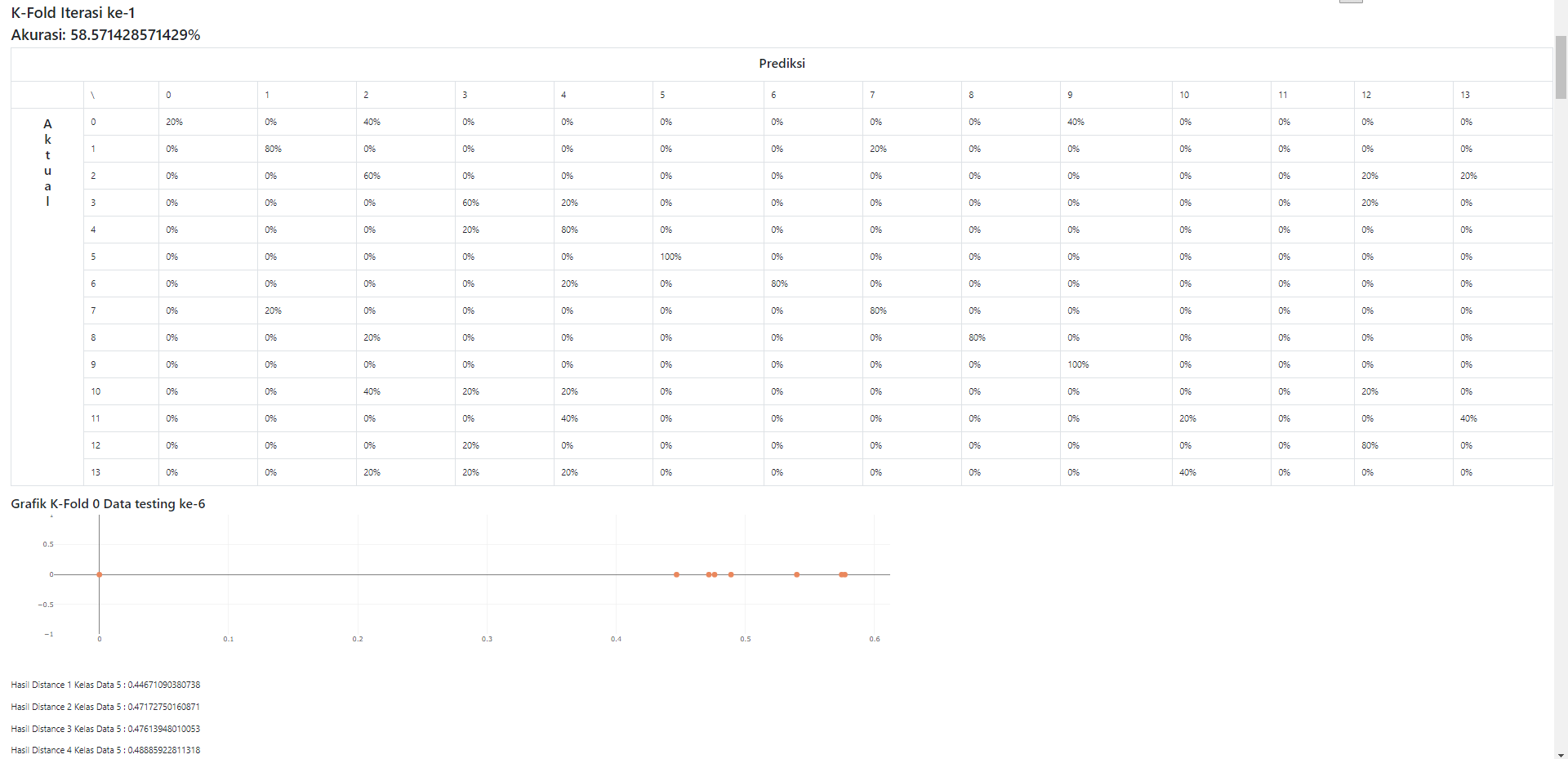
**Lampiran B.3.** Implementasi Sistem pada KNN dan MKNN Pengujian Pertama

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). Form dari input nilai *k* seperti pada Gambar 1.



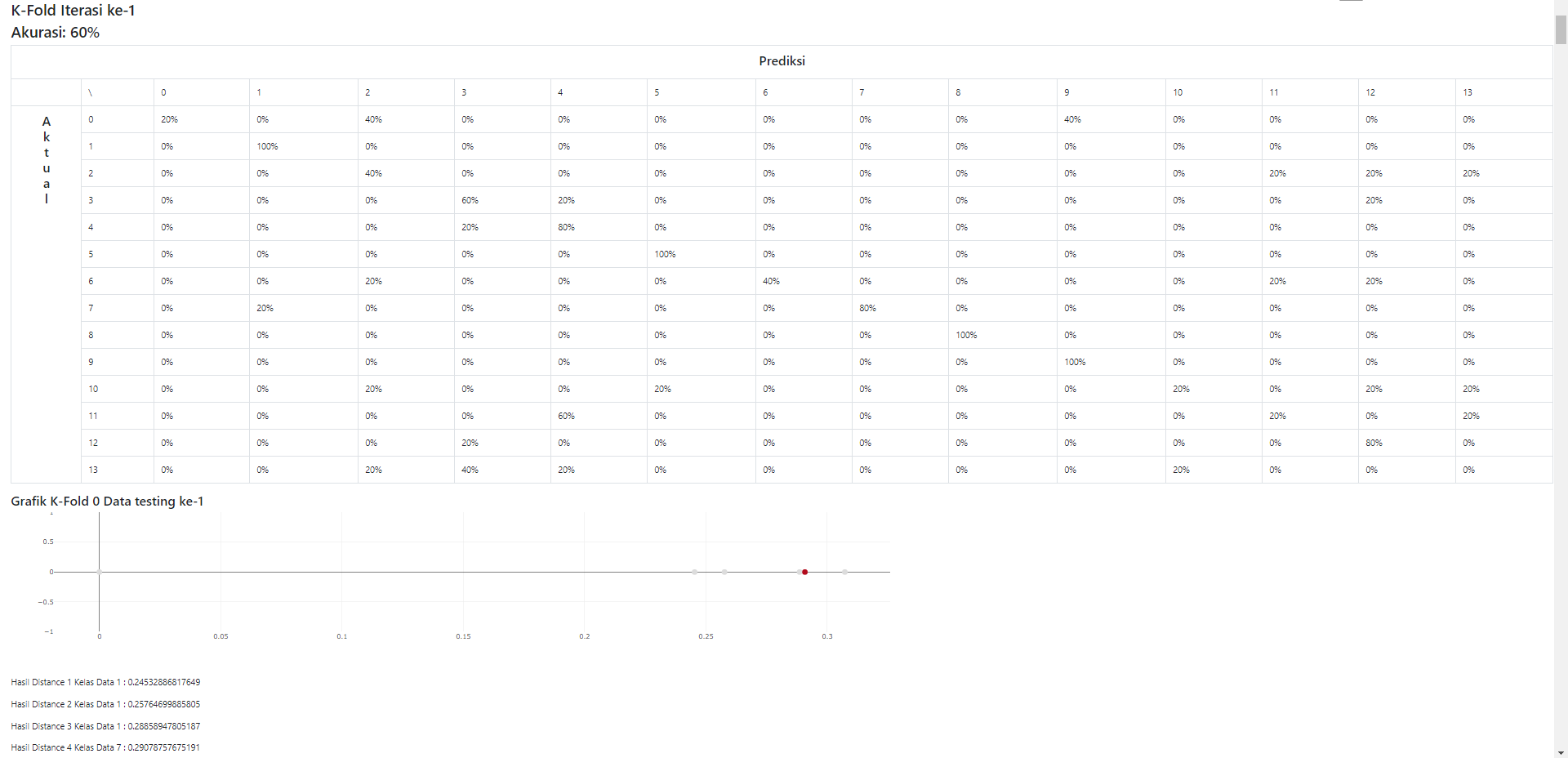
Gambar 1. Input Nilai k

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



Gambar 2. Output Klasifikasi KNN

Output dari MKNN akan munjul jika nilai *k* telah diinputkan. Tampilan pada Gambar 3, merupakan hasil dari klasifikasi MKNN.



Gambar 3. Output 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 | 2 | -0.0091 |
| 2 | 1 | -1.3658 |
| 3 | 4 | 0.29293 |
| 4 | 1 | -1.3266 |
| 5 | 6 | 0.29202 |
| 6 | 4 | 0.58009 |
| 7 | 5 | 0.29897 |
| 8 | 5 | 0.71151 |
| 9 | 3 | 0.3556 |
| 10 | 5 | 0.21344 |
| 11 | 13 | 0.40506 |
| 12 | 12 | 0.41782 |
| 13 | 8 | 0.06721 |
| 14 | 0 | -0.3793 |
| 15 | 11 | 0.6933 |
| 16 | 5 | -0.2461 |
| 17 | 6 | 0.55632 |
| 18 | 0 | -0.155 |
| 19 | 0 | -0.6014 |
| 20 | 2 | -0.1276 |
| 21 | 1 | -1.2845 |
| 22 | 4 | 0.05549 |
| 23 | 10 | 0.53527 |
| 24 | 10 | 0.16187 |
| 25 | 5 | 0.3328 |
| 26 | 2 | 0.54624 |
| 27 | 11 | 0.18314 |
| 28 | 0 | -0.1791 |
| 29 | 7 | -0.9654 |
| 30 | 2 | -0.0045 |
| 31 | 12 | -0.0097 |
| 32 | 0 | -0.2065 |
| 33 | 8 | 0.37701 |
| 34 | 13 | 0.27642 |
| 35 | 3 | 0.34585 |
| 36 | 2 | -0.2774 |
| 37 | 6 | 0.07976 |
| 38 | 9 | -0.6192 |
| 39 | 12 | 0.61952 |
| 40 | 8 | 0.32183 |
| 41 | 6 | 0.53986 |
| 42 | 13 | 0.57015 |
| 43 | 1 | -1.3899 |
| 44 | 4 | 0.50801 |
| 45 | 11 | 0.52067 |
| 46 | 4 | 0.28339 |
| 47 | 1 | -0.9152 |
| 48 | 0 | -0.0682 |
| 49 | 0 | -0.1074 |
| 50 | 3 | 0.55495 |
| 51 | 5 | 0.2732 |
| 52 | 7 | -1.3544 |
| 53 | 2 | -0.4262 |
| 54 | 3 | 0.18595 |
| 55 | 13 | 0.56233 |
| 56 | 0 | -0.5163 |
| 57 | 13 | 0.07233 |
| 58 | 6 | 0.32191 |
| 59 | 13 | 0.12885 |
| 60 | 5 | 0.01771 |
| 61 | 8 | 0.59039 |
| 62 | 4 | 0.58563 |
| 63 | 9 | -0.7256 |
| 64 | 3 | 0.30912 |
| 65 | 9 | -0.5266 |
| 66 | 4 | 0.3601 |
| 67 | 4 | 0.24993 |
| 68 | 9 | -0.6475 |
| 69 | 2 | 0.00363 |
| 70 | 9 | -0.4402 |
| 71 | 11 | 0.18066 |
| 72 | 11 | 0.65118 |
| 73 | 7 | -1.3577 |
| 74 | 7 | -0.8608 |
| 75 | 12 | 0.46947 |
| 76 | 3 | 0.23401 |
| 77 | 2 | -0.5713 |
| 78 | 9 | -0.5974 |
| 79 | 12 | 0.53978 |
| 80 | 1 | -0.8903 |
| 81 | 2 | 0.29067 |
| 82 | 12 | 0.37103 |
| 83 | 3 | 0.19678 |
| 84 | 10 | -0.1118 |
| 85 | 2 | 0.12251 |
| 86 | 13 | 0.16322 |
| 87 | 6 | 0.36675 |
| 88 | 9 | -0.5323 |
| 89 | 0 | -0.2051 |
| 90 | 8 | 0.45093 |
| 91 | 11 | 0.46221 |
| 92 | 7 | -0.8762 |
| 93 | 1 | -1.2488 |
| 94 | 7 | -0.9279 |
| 95 | 11 | -0.4082 |
| 96 | 13 | -0.1277 |
| 97 | 11 | 0.26866 |
| 98 | 10 | 0.13333 |
| 99 | 2 | 0.01009 |
| 100 | 6 | 0.62453 |
| - | - | - |
| - | - | - |
| - | - | - |
| - | - | - |
| 700 | 1 | -1.1201 |

1. Reduksi menggunakan PCA dengan 2 variabel fitur baru.

|  |  |  |  |
| --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 |
| 1 | 2 | -0.0091 | -0.3742 |
| 2 | 1 | -1.3658 | -0.0787 |
| 3 | 4 | 0.29293 | -0.4173 |
| 4 | 1 | -1.3266 | -0.1404 |
| 5 | 6 | 0.29202 | -0.5083 |
| 6 | 4 | 0.58009 | 0.59651 |
| 7 | 5 | 0.29897 | -0.4633 |
| 8 | 5 | 0.71151 | 0.46473 |
| 9 | 3 | 0.3556 | -0.4967 |
| 10 | 5 | 0.21344 | -0.3884 |
| 11 | 13 | 0.40506 | -0.5186 |
| 12 | 12 | 0.41782 | -0.5281 |
| 13 | 8 | 0.06721 | -0.3101 |
| 14 | 0 | -0.3793 | 0.6998 |
| 15 | 11 | 0.6933 | 0.57028 |
| 16 | 5 | -0.2461 | -0.4345 |
| 17 | 6 | 0.55632 | 0.601 |
| 18 | 0 | -0.155 | -0.2606 |
| 19 | 0 | -0.6014 | 0.70943 |
| 20 | 2 | -0.1276 | -0.4503 |
| 21 | 1 | -1.2845 | -0.0573 |
| 22 | 4 | 0.05549 | -0.4698 |
| 23 | 10 | 0.53527 | 0.52163 |
| 24 | 10 | 0.16187 | -0.5085 |
| 25 | 5 | 0.3328 | -0.4264 |
| 26 | 2 | 0.54624 | 0.51219 |
| 27 | 11 | 0.18314 | -0.4863 |
| 28 | 0 | -0.1791 | -0.4115 |
| 29 | 7 | -0.9654 | 0.80724 |
| 30 | 2 | -0.0045 | -0.4109 |
| 31 | 12 | -0.0097 | -0.3891 |
| 32 | 0 | -0.2065 | 0.67226 |
| 33 | 8 | 0.37701 | -0.4322 |
| 34 | 13 | 0.27642 | -0.4933 |
| 35 | 3 | 0.34585 | -0.508 |
| 36 | 2 | -0.2774 | -0.3393 |
| 37 | 6 | 0.07976 | -0.3512 |
| 38 | 9 | -0.6192 | -0.3246 |
| 39 | 12 | 0.61952 | 0.43818 |
| 40 | 8 | 0.32183 | -0.4778 |
| 41 | 6 | 0.53986 | 0.47038 |
| 42 | 13 | 0.57015 | 0.49109 |
| 43 | 1 | -1.3899 | -0.0961 |
| 44 | 4 | 0.50801 | -0.5529 |
| 45 | 11 | 0.52067 | 0.53077 |
| 46 | 4 | 0.28339 | -0.3786 |
| 47 | 1 | -0.9152 | 0.85401 |
| 48 | 0 | -0.0682 | 0.56654 |
| 49 | 0 | -0.1074 | 0.5659 |
| 50 | 3 | 0.55495 | 0.53179 |
| 51 | 5 | 0.2732 | -0.4157 |
| 52 | 7 | -1.3544 | -0.1587 |
| 53 | 2 | -0.4262 | -0.2367 |
| 54 | 3 | 0.18595 | -0.4945 |
| 55 | 13 | 0.56233 | 0.416 |
| 56 | 0 | -0.5163 | 0.79405 |
| 57 | 13 | 0.07233 | -0.3545 |
| 58 | 6 | 0.32191 | -0.4189 |
| 59 | 13 | 0.12885 | -0.3711 |
| 60 | 5 | 0.01771 | -0.3843 |
| 61 | 8 | 0.59039 | 0.54937 |
| 62 | 4 | 0.58563 | 0.58581 |
| 63 | 9 | -0.7256 | -0.3249 |
| 64 | 3 | 0.30912 | 0.48059 |
| 65 | 9 | -0.5266 | -0.2257 |
| 66 | 4 | 0.3601 | -0.5413 |
| 67 | 4 | 0.24993 | -0.4904 |
| 68 | 9 | -0.6475 | -0.2244 |
| 69 | 2 | 0.00363 | -0.3416 |
| 70 | 9 | -0.4402 | -0.3552 |
| 71 | 11 | 0.18066 | -0.3436 |
| 72 | 11 | 0.65118 | 0.3944 |
| 73 | 7 | -1.3577 | -0.1472 |
| 74 | 7 | -0.8608 | 0.80326 |
| 75 | 12 | 0.46947 | 0.48715 |
| 76 | 3 | 0.23401 | -0.4187 |
| 77 | 2 | -0.5713 | -0.365 |
| 78 | 9 | -0.5974 | -0.2438 |
| 79 | 12 | 0.53978 | 0.5439 |
| 80 | 1 | -0.8903 | 0.8734 |
| 81 | 2 | 0.29067 | -0.3725 |
| 82 | 12 | 0.37103 | 0.66894 |
| 83 | 3 | 0.19678 | 0.67594 |
| 84 | 10 | -0.1118 | -0.3192 |
| 85 | 2 | 0.12251 | -0.3481 |
| 86 | 13 | 0.16322 | -0.473 |
| 87 | 6 | 0.36675 | -0.4069 |
| 88 | 9 | -0.5323 | -0.35 |
| 89 | 0 | -0.2051 | 0.62893 |
| 90 | 8 | 0.45093 | -0.4135 |
| 91 | 11 | 0.46221 | -0.4492 |
| 92 | 7 | -0.8762 | 0.8638 |
| 93 | 1 | -1.2488 | -0.0982 |
| 94 | 7 | -0.9279 | -0.1864 |
| 95 | 11 | -0.4082 | -0.3456 |
| 96 | 13 | -0.1277 | -0.4331 |
| 97 | 11 | 0.26866 | 0.64093 |
| 98 | 10 | 0.13333 | -0.4752 |
| 99 | 2 | 0.01009 | -0.2853 |
| 100 | 6 | 0.62453 | 0.42892 |
| - | - | - | - |
| - | - | - | - |
| - | - | - | - |
| - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 |

1. Reduksi menggunakan PCA dengan 3 variabel fitur baru.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 |
| - | - | - | - | - |
| - | - | - | - | - |
| - | - | - | - | - |
| - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 |

1. Reduksi menggunakan PCA dengan 4 variabel fitur baru.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 |

1. Reduksi menggunakan PCA dengan 5 variabel fitur baru.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.1485 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.15139 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.12882 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.0349 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.13211 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.2581 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.1338 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.03254 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.2193 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.12329 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.11709 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.61397 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.374 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.08442 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.5235 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.0695 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.236 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.6474 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.1388 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.26724 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.67386 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.01152 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.0097 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.2376 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.1658 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.0423 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.1996 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.23135 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.088 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.2465 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.16617 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.03324 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.04454 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.2045 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.046 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.432 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.0484 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.08678 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.02999 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.02002 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.1727 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.19726 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.00207 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.10792 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.23899 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.3405 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.3097 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.0274 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.2375 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.18021 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.3267 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.0652 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.11148 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.3714 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.096 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.0143 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.0298 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.3396 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.05681 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.5251 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.55514 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.4185 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.22991 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.06381 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.4964 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.1124 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.3331 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.054 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.10501 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.19588 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.31848 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.0291 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.00435 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.407 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.4485 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4E-05 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.22661 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.05056 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.14657 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.80974 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.0348 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.01873 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.15039 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.4131 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.23372 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.04914 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.16407 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.24477 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.22877 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.40488 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.22541 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.1559 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.118 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.2665 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.104 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 |
| - | - | - | - | - | - | - |
| - | - | - | - | - | - | - |
| - | - | - | - | - | - | - |
| - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.32437 |

1. Reduksi menggunakan PCA dengan 6 variabel fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 | pca6 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.1485 | 0.06512 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.15139 | -0.0909 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 | -0.0495 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.12882 | -0.1145 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.0349 | 0.01449 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.13211 | -0.0549 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.2581 | -0.4956 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.1338 | -0.7184 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.03254 | 0.03703 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.2193 | -0.5723 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.12329 | 0.00601 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.11709 | 0.02547 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.61397 | 0.19328 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.374 | 0.24264 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.08442 | -0.2526 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.5235 | -0.6602 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.0695 | 0.05176 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.236 | -0.0768 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.6474 | 0.01464 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.1388 | -0.0235 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.26724 | -0.1411 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.67386 | -0.08 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.01152 | 0.07098 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.0097 | -0.0376 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.2376 | -0.5342 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.1658 | -0.3223 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.0423 | -0.0248 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.1996 | 0.14409 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.23135 | 0.01893 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.088 | 0.39082 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 | -0.0823 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.2465 | 0.01659 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.16617 | 0.33837 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.03324 | -0.0133 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.04454 | 0.08697 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.2045 | 0.05315 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.046 | 0.09332 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.432 | 0.13571 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.0484 | -0.0365 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.08678 | 0.37374 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.02999 | 0.05604 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.02002 | -0.1481 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.1727 | -0.0483 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.19726 | 0.02498 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.00207 | -0.1106 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.10792 | 0.00302 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.23899 | -0.0492 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.3405 | 0.0268 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.3097 | 0.26271 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.0274 | 0.02059 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.2375 | -0.6533 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.18021 | -0.156 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.3267 | 0.11546 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.0652 | -0.1179 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.11148 | 0.00432 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.3714 | -0.2546 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.096 | -0.0243 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.0143 | -0.0738 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.0298 | -0.1117 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.3396 | -0.7068 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.05681 | 0.46601 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 | 0.15947 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.5251 | 0.26576 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.55514 | -0.7534 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.4185 | 0.19652 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.22991 | 0.01815 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.06381 | -0.0232 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.4964 | 0.12258 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.1124 | 0.2255 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.3331 | 0.1805 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.054 | -0.0439 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.10501 | 0.06596 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.19588 | -0.1193 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.31848 | -0.0371 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.0291 | -0.0615 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.00435 | -0.0958 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.407 | 0.30086 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.4485 | 0.09305 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4E-05 | 0.03077 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.22661 | -0.0053 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.05056 | 0.01889 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.14657 | 0.02527 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.80974 | -0.1262 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14 | 0.11926 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.0348 | -0.0066 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.01873 | 0.00108 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.15039 | 0.27929 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.4131 | 0.18879 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.23372 | -0.0124 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.04914 | 0.19679 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.16407 | -0.0312 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.24477 | 0.1822 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.22877 | -0.1288 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.40488 | 2E-05 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.22541 | -0.1229 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.1559 | 0.05741 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.118 | -0.0972 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.2665 | -0.3425 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.104 | 0.26264 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 | -0.535 |
| - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.32437 | -0.1438 |

1. Reduksi menggunakan PCA dengan 7 variabel fitur baru.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 | pca6 | pca7 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.1485 | 0.06512 | 0.32701 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.15139 | -0.0909 | 0.2377 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 | -0.0495 | -0.0394 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.12882 | -0.1145 | 0.20513 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.0349 | 0.01449 | -0.0284 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.13211 | -0.0549 | 0.0289 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.2581 | -0.4956 | 0.13563 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.1338 | -0.7184 | -0.184 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.03254 | 0.03703 | 0.25735 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.2193 | -0.5723 | -0.1602 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.12329 | 0.00601 | 0.01869 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.11709 | 0.02547 | 0.03878 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.61397 | 0.19328 | -0.0241 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.374 | 0.24264 | -0.0495 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.08442 | -0.2526 | -0.0574 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.5235 | -0.6602 | -0.2713 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.0695 | 0.05176 | 0.15986 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.236 | -0.0768 | 0.26122 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.6474 | 0.01464 | 0.08828 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.1388 | -0.0235 | -0.1094 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.26724 | -0.1411 | 0.13439 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.67386 | -0.08 | -0.1587 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.01152 | 0.07098 | 0.18941 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.0097 | -0.0376 | -0.168 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.2376 | -0.5342 | 0.17538 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.1658 | -0.3223 | 0.13974 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.0423 | -0.0248 | -0.1836 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.1996 | 0.14409 | 0.14834 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.23135 | 0.01893 | -0.1237 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.088 | 0.39082 | -0.3415 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 | -0.0823 | 0.26149 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.2465 | 0.01659 | 0.10568 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.16617 | 0.33837 | -0.2849 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.03324 | -0.0133 | -0.0615 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.04454 | 0.08697 | 0.27031 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.2045 | 0.05315 | -0.2881 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.046 | 0.09332 | -0.0876 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.432 | 0.13571 | -0.113 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.0484 | -0.0365 | 0.04372 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.08678 | 0.37374 | 0.01626 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.02999 | 0.05604 | -0.1649 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.02002 | -0.1481 | -0.1944 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.1727 | -0.0483 | 0.2335 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.19726 | 0.02498 | -0.0312 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.00207 | -0.1106 | 0.18488 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.10792 | 0.00302 | 0.04879 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.23899 | -0.0492 | -0.0236 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.3405 | 0.0268 | 0.393 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.3097 | 0.26271 | -0.137 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.0274 | 0.02059 | 0.19845 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.2375 | -0.6533 | 0.24419 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.18021 | -0.156 | 0.22465 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.3267 | 0.11546 | -0.0818 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.0652 | -0.1179 | -0.1422 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.11148 | 0.00432 | -0.0707 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.3714 | -0.2546 | -0.0939 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.096 | -0.0243 | 0.30465 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.0143 | -0.0738 | -0.1356 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.0298 | -0.1117 | -0.1828 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.3396 | -0.7068 | -0.2048 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.05681 | 0.46601 | -0.0148 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 | 0.15947 | -0.1745 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.5251 | 0.26576 | -0.1393 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.55514 | -0.7534 | -0.2962 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.4185 | 0.19652 | -0.111 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.22991 | 0.01815 | -0.0699 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.06381 | -0.0232 | 0.041 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.4964 | 0.12258 | 0.11779 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.1124 | 0.2255 | 0.38338 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.3331 | 0.1805 | -0.308 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.054 | -0.0439 | 0.07623 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.10501 | 0.06596 | -0.2602 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.19588 | -0.1193 | 0.24184 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.31848 | -0.0371 | -0.2462 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.0291 | -0.0615 | 0.34582 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.00435 | -0.0958 | 0.25759 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.407 | 0.30086 | -0.4387 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.4485 | 0.09305 | -0.0214 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4E-05 | 0.03077 | 0.20145 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.22661 | -0.0053 | -0.03 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.05056 | 0.01889 | 0.20309 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.14657 | 0.02527 | 0.40505 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.80974 | -0.1262 | 0.04039 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14 | 0.11926 | 0.01305 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.0348 | -0.0066 | 0.19462 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.01873 | 0.00108 | -0.0932 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.15039 | 0.27929 | -0.1926 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.4131 | 0.18879 | -0.2344 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.23372 | -0.0124 | 0.20339 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.04914 | 0.19679 | -0.1033 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.16407 | -0.0312 | -0.0739 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.24477 | 0.1822 | -0.2571 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.22877 | -0.1288 | 0.13157 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.40488 | 2E-05 | -0.442 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.22541 | -0.1229 | 0.29052 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.1559 | 0.05741 | -0.1247 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.118 | -0.0972 | -0.5592 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.2665 | -0.3425 | 0.2802 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.104 | 0.26264 | -0.2354 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 | -0.535 | 0.05176 |
| - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.32437 | -0.1438 | -0.093 |

1. Reduksi menggunakan PCA dengan 8 variabel fitur baru.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 | pca6 | pca7 | pca8 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.1485 | 0.06512 | 0.32701 | -0.0248 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.15139 | -0.0909 | 0.2377 | 0.09373 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 | -0.0495 | -0.0394 | -0.0544 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.12882 | -0.1145 | 0.20513 | -0.0112 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.0349 | 0.01449 | -0.0284 | -0.1438 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.13211 | -0.0549 | 0.0289 | 0.13353 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.2581 | -0.4956 | 0.13563 | -0.0115 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.1338 | -0.7184 | -0.184 | -0.0156 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.03254 | 0.03703 | 0.25735 | -0.0238 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.2193 | -0.5723 | -0.1602 | -0.1337 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.12329 | 0.00601 | 0.01869 | 0.15042 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.11709 | 0.02547 | 0.03878 | 0.24211 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.61397 | 0.19328 | -0.0241 | 0.00837 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.374 | 0.24264 | -0.0495 | -0.122 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.08442 | -0.2526 | -0.0574 | 0.08785 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.5235 | -0.6602 | -0.2713 | 0.01492 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.0695 | 0.05176 | 0.15986 | -0.1353 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.236 | -0.0768 | 0.26122 | 0.08589 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.6474 | 0.01464 | 0.08828 | 0.01728 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.1388 | -0.0235 | -0.1094 | 0.16811 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.26724 | -0.1411 | 0.13439 | -0.0375 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.67386 | -0.08 | -0.1587 | -0.1066 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.01152 | 0.07098 | 0.18941 | 0.08956 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.0097 | -0.0376 | -0.168 | 0.11341 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.2376 | -0.5342 | 0.17538 | -0.0279 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.1658 | -0.3223 | 0.13974 | -0.0418 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.0423 | -0.0248 | -0.1836 | -0.3593 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.1996 | 0.14409 | 0.14834 | -0.2077 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.23135 | 0.01893 | -0.1237 | -0.1841 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.088 | 0.39082 | -0.3415 | -0.1676 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 | -0.0823 | 0.26149 | 0.32333 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.2465 | 0.01659 | 0.10568 | -0.3655 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.16617 | 0.33837 | -0.2849 | -0.046 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.03324 | -0.0133 | -0.0615 | 0.12689 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.04454 | 0.08697 | 0.27031 | 0.02133 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.2045 | 0.05315 | -0.2881 | 0.17749 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.046 | 0.09332 | -0.0876 | 0.1184 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.432 | 0.13571 | -0.113 | 0.17215 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.0484 | -0.0365 | 0.04372 | 0.13451 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.08678 | 0.37374 | 0.01626 | 0.103 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.02999 | 0.05604 | -0.1649 | -0.028 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.02002 | -0.1481 | -0.1944 | 0.09352 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.1727 | -0.0483 | 0.2335 | -0.1161 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.19726 | 0.02498 | -0.0312 | 0.18332 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.00207 | -0.1106 | 0.18488 | -0.0717 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.10792 | 0.00302 | 0.04879 | -0.0084 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.23899 | -0.0492 | -0.0236 | 0.1183 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.3405 | 0.0268 | 0.393 | -0.0805 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.3097 | 0.26271 | -0.137 | 0.01677 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.0274 | 0.02059 | 0.19845 | 0.10062 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.2375 | -0.6533 | 0.24419 | -0.0206 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.18021 | -0.156 | 0.22465 | -0.0888 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.3267 | 0.11546 | -0.0818 | 0.0943 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.0652 | -0.1179 | -0.1422 | 0.17143 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.11148 | 0.00432 | -0.0707 | -0.0017 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.3714 | -0.2546 | -0.0939 | 0.16392 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.096 | -0.0243 | 0.30465 | 0.16925 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.0143 | -0.0738 | -0.1356 | 0.00371 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.0298 | -0.1117 | -0.1828 | 0.22396 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.3396 | -0.7068 | -0.2048 | 0.13615 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.05681 | 0.46601 | -0.0148 | 0.07734 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 | 0.15947 | -0.1745 | -0.1905 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.5251 | 0.26576 | -0.1393 | -0.0592 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.55514 | -0.7534 | -0.2962 | -0.1681 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.4185 | 0.19652 | -0.111 | 0.16053 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.22991 | 0.01815 | -0.0699 | 0.10827 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.06381 | -0.0232 | 0.041 | -0.0442 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.4964 | 0.12258 | 0.11779 | 0.32366 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.1124 | 0.2255 | 0.38338 | -0.2175 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.3331 | 0.1805 | -0.308 | 0.02607 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.054 | -0.0439 | 0.07623 | 0.14068 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.10501 | 0.06596 | -0.2602 | 0.0165 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.19588 | -0.1193 | 0.24184 | 0.09257 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.31848 | -0.0371 | -0.2462 | 0.00466 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.0291 | -0.0615 | 0.34582 | 0.23521 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.00435 | -0.0958 | 0.25759 | -0.0452 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.407 | 0.30086 | -0.4387 | 0.00074 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.4485 | 0.09305 | -0.0214 | 0.2387 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4E-05 | 0.03077 | 0.20145 | 0.31122 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.22661 | -0.0053 | -0.03 | 0.07077 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.05056 | 0.01889 | 0.20309 | 0.09507 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.14657 | 0.02527 | 0.40505 | 0.36004 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.80974 | -0.1262 | 0.04039 | 0.00603 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14 | 0.11926 | 0.01305 | -0.083 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.0348 | -0.0066 | 0.19462 | 0.31315 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.01873 | 0.00108 | -0.0932 | 0.12348 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.15039 | 0.27929 | -0.1926 | -0.413 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.4131 | 0.18879 | -0.2344 | 0.08656 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.23372 | -0.0124 | 0.20339 | 0.0392 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.04914 | 0.19679 | -0.1033 | 0.06512 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.16407 | -0.0312 | -0.0739 | -0.0514 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.24477 | 0.1822 | -0.2571 | -0.0858 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.22877 | -0.1288 | 0.13157 | -0.0425 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.40488 | 2E-05 | -0.442 | -0.1953 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.22541 | -0.1229 | 0.29052 | -0.09 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.1559 | 0.05741 | -0.1247 | 0.02238 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.118 | -0.0972 | -0.5592 | 0.07689 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.2665 | -0.3425 | 0.2802 | -0.1693 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.104 | 0.26264 | -0.2354 | 0.22795 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 | -0.535 | 0.05176 | -0.1408 |
| - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.32437 | -0.1438 | -0.093 | 0.18783 |

1. Reduksi menggunakan PCA dengan 9 variabel fitur baru.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 | pca6 | pca7 | pca8 | pca9 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.14848 | 0.06512 | 0.32701 | -0.0248 | 0.00552 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.151394 | -0.0909 | 0.2377 | 0.09373 | -0.0854 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 | -0.0495 | -0.0394 | -0.0544 | -0.2193 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.128823 | -0.1145 | 0.20513 | -0.0112 | -0.0489 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.03491 | 0.01449 | -0.0284 | -0.1438 | 0.20002 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.132108 | -0.0549 | 0.0289 | 0.13353 | -0.1019 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.25805 | -0.4956 | 0.13563 | -0.0115 | -0.2616 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.13376 | -0.7184 | -0.184 | -0.0156 | -0.0183 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.032545 | 0.03703 | 0.25735 | -0.0238 | 0.19603 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.21932 | -0.5723 | -0.1602 | -0.1337 | -0.0537 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.123293 | 0.00601 | 0.01869 | 0.15042 | 0.00265 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.117089 | 0.02547 | 0.03878 | 0.24211 | -0.1473 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.613971 | 0.19328 | -0.0241 | 0.00837 | 0.0274 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.37398 | 0.24264 | -0.0495 | -0.122 | -0.0195 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.084423 | -0.2526 | -0.0574 | 0.08785 | 0.04424 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.52346 | -0.6602 | -0.2713 | 0.01492 | 0.16259 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.06955 | 0.05176 | 0.15986 | -0.1353 | 0.2129 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.23596 | -0.0768 | 0.26122 | 0.08589 | -0.1485 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.64742 | 0.01464 | 0.08828 | 0.01728 | 0.06802 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.13878 | -0.0235 | -0.1094 | 0.16811 | -0.0266 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.267237 | -0.1411 | 0.13439 | -0.0375 | -0.0209 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.673863 | -0.08 | -0.1587 | -0.1066 | -0.0151 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.011519 | 0.07098 | 0.18941 | 0.08956 | -0.0093 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.00967 | -0.0376 | -0.168 | 0.11341 | -0.0319 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.23761 | -0.5342 | 0.17538 | -0.0279 | 0.03791 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.16582 | -0.3223 | 0.13974 | -0.0418 | 0.09719 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.04229 | -0.0248 | -0.1836 | -0.3593 | 0.06465 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.19957 | 0.14409 | 0.14834 | -0.2077 | -0.2032 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.231354 | 0.01893 | -0.1237 | -0.1841 | 0.09792 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.08801 | 0.39082 | -0.3415 | -0.1676 | -0.0693 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 | -0.0823 | 0.26149 | 0.32333 | -0.0214 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.24651 | 0.01659 | 0.10568 | -0.3655 | -0.3521 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.166169 | 0.33837 | -0.2849 | -0.046 | -0.2973 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.033237 | -0.0133 | -0.0615 | 0.12689 | 0.04979 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.044543 | 0.08697 | 0.27031 | 0.02133 | 0.04598 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.20454 | 0.05315 | -0.2881 | 0.17749 | 0.05195 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.04596 | 0.09332 | -0.0876 | 0.1184 | 0.19621 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.43195 | 0.13571 | -0.113 | 0.17215 | 0.01917 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.04842 | -0.0365 | 0.04372 | 0.13451 | 0.04975 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.086781 | 0.37374 | 0.01626 | 0.103 | 0.019 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.029991 | 0.05604 | -0.1649 | -0.028 | 0.12594 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.020019 | -0.1481 | -0.1944 | 0.09352 | 0.00926 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.172705 | -0.0483 | 0.2335 | -0.1161 | -0.0874 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.197258 | 0.02498 | -0.0312 | 0.18332 | -0.0548 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.002073 | -0.1106 | 0.18488 | -0.0717 | 0.07699 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.107917 | 0.00302 | 0.04879 | -0.0084 | -0.2589 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.238985 | -0.0492 | -0.0236 | 0.1183 | 0.20592 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.34047 | 0.0268 | 0.393 | -0.0805 | -0.1467 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.30969 | 0.26271 | -0.137 | 0.01677 | -0.266 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.02744 | 0.02059 | 0.19845 | 0.10062 | 0.17456 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.23754 | -0.6533 | 0.24419 | -0.0206 | -0.1346 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.180208 | -0.156 | 0.22465 | -0.0888 | 0.12955 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.32671 | 0.11546 | -0.0818 | 0.0943 | 0.02337 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.06523 | -0.1179 | -0.1422 | 0.17143 | 0.11165 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.111476 | 0.00432 | -0.0707 | -0.0017 | -0.0407 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.37136 | -0.2546 | -0.0939 | 0.16392 | 0.06805 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.09599 | -0.0243 | 0.30465 | 0.16925 | -0.0355 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.01427 | -0.0738 | -0.1356 | 0.00371 | 0.12441 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.02981 | -0.1117 | -0.1828 | 0.22396 | 0.00595 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.33957 | -0.7068 | -0.2048 | 0.13615 | 0.01817 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.056807 | 0.46601 | -0.0148 | 0.07734 | 0.01795 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 | 0.15947 | -0.1745 | -0.1905 | -0.1442 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.52512 | 0.26576 | -0.1393 | -0.0592 | 0.02158 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.555138 | -0.7534 | -0.2962 | -0.1681 | 0.33885 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.41847 | 0.19652 | -0.111 | 0.16053 | 0.00337 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.229913 | 0.01815 | -0.0699 | 0.10827 | -0.0335 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.063811 | -0.0232 | 0.041 | -0.0442 | -0.1894 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.49642 | 0.12258 | 0.11779 | 0.32366 | 0.09255 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.11244 | 0.2255 | 0.38338 | -0.2175 | -0.1109 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.33308 | 0.1805 | -0.308 | 0.02607 | 0.03676 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.053998 | -0.0439 | 0.07623 | 0.14068 | -0.0486 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.105006 | 0.06596 | -0.2602 | 0.0165 | -0.0574 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.195881 | -0.1193 | 0.24184 | 0.09257 | -0.0382 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.318482 | -0.0371 | -0.2462 | 0.00466 | -0.0194 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.02908 | -0.0615 | 0.34582 | 0.23521 | 0.00584 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.004353 | -0.0958 | 0.25759 | -0.0452 | 0.14724 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.40702 | 0.30086 | -0.4387 | 0.00074 | -0.0289 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.44847 | 0.09305 | -0.0214 | 0.2387 | 0.00506 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4.40E-05 | 0.03077 | 0.20145 | 0.31122 | -0.0331 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.226611 | -0.0053 | -0.03 | 0.07077 | -0.12 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.050563 | 0.01889 | 0.20309 | 0.09507 | -0.0566 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.146571 | 0.02527 | 0.40505 | 0.36004 | 0.00626 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.809738 | -0.1262 | 0.04039 | 0.00603 | 0.11497 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14004 | 0.11926 | 0.01305 | -0.083 | -0.0094 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.03485 | -0.0066 | 0.19462 | 0.31315 | -0.0524 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.018735 | 0.00108 | -0.0932 | 0.12348 | -0.1179 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.150392 | 0.27929 | -0.1926 | -0.413 | 0.14339 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.41315 | 0.18879 | -0.2344 | 0.08656 | 0.12136 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.233725 | -0.0124 | 0.20339 | 0.0392 | -0.0976 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.049141 | 0.19679 | -0.1033 | 0.06512 | -0.0786 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.164068 | -0.0312 | -0.0739 | -0.0514 | -0.257 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.244765 | 0.1822 | -0.2571 | -0.0858 | 0.00383 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.228765 | -0.1288 | 0.13157 | -0.0425 | -0.0489 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.404881 | 1.95E-05 | -0.442 | -0.1953 | -0.0199 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.225409 | -0.1229 | 0.29052 | -0.09 | 0.1076 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.15587 | 0.05741 | -0.1247 | 0.02238 | 0.00186 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.11801 | -0.0972 | -0.5592 | 0.07689 | 0.04147 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.26652 | -0.3425 | 0.2802 | -0.1693 | 0.17647 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.10396 | 0.26264 | -0.2354 | 0.22795 | 0.017 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 | -0.535 | 0.05176 | -0.1408 | 0.24745 |
| - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.324374 | -0.1438 | -0.093 | 0.18783 | -0.0205 |

1. Reduksi menggunakan PCA dengan 10 variabel fitur baru.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 | pca6 | pca7 | pca8 | pca9 | pca10 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.1485 | 0.06512 | 0.32701 | -0.0248 | 0.00552 | 0.0637 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.15139 | -0.0909 | 0.2377 | 0.09373 | -0.0854 | 0.17142 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 | -0.0495 | -0.0394 | -0.0544 | -0.2193 | -0.0462 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.12882 | -0.1145 | 0.20513 | -0.0112 | -0.0489 | 0.00561 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.0349 | 0.01449 | -0.0284 | -0.1438 | 0.20002 | 0.22582 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.13211 | -0.0549 | 0.0289 | 0.13353 | -0.1019 | 0.14967 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.2581 | -0.4956 | 0.13563 | -0.0115 | -0.2616 | -0.1396 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.1338 | -0.7184 | -0.184 | -0.0156 | -0.0183 | -0.1347 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.03254 | 0.03703 | 0.25735 | -0.0238 | 0.19603 | -0.1432 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.2193 | -0.5723 | -0.1602 | -0.1337 | -0.0537 | -0.0685 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.12329 | 0.00601 | 0.01869 | 0.15042 | 0.00265 | 0.05228 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.11709 | 0.02547 | 0.03878 | 0.24211 | -0.1473 | 0.06039 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.61397 | 0.19328 | -0.0241 | 0.00837 | 0.0274 | 0.00227 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.374 | 0.24264 | -0.0495 | -0.122 | -0.0195 | -0.1481 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.08442 | -0.2526 | -0.0574 | 0.08785 | 0.04424 | -0.0333 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.5235 | -0.6602 | -0.2713 | 0.01492 | 0.16259 | 0.01595 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.0695 | 0.05176 | 0.15986 | -0.1353 | 0.2129 | 0.01113 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.236 | -0.0768 | 0.26122 | 0.08589 | -0.1485 | 0.14805 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.6474 | 0.01464 | 0.08828 | 0.01728 | 0.06802 | 0.04276 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.1388 | -0.0235 | -0.1094 | 0.16811 | -0.0266 | 0.15636 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.26724 | -0.1411 | 0.13439 | -0.0375 | -0.0209 | 0.13199 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.67386 | -0.08 | -0.1587 | -0.1066 | -0.0151 | 0.01428 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.01152 | 0.07098 | 0.18941 | 0.08956 | -0.0093 | -0.0593 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.0097 | -0.0376 | -0.168 | 0.11341 | -0.0319 | -0.018 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.2376 | -0.5342 | 0.17538 | -0.0279 | 0.03791 | -0.1151 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.1658 | -0.3223 | 0.13974 | -0.0418 | 0.09719 | -0.0006 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.0423 | -0.0248 | -0.1836 | -0.3593 | 0.06465 | 0.02257 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.1996 | 0.14409 | 0.14834 | -0.2077 | -0.2032 | -0.0632 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.23135 | 0.01893 | -0.1237 | -0.1841 | 0.09792 | -0.001 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.088 | 0.39082 | -0.3415 | -0.1676 | -0.0693 | -0.0613 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 | -0.0823 | 0.26149 | 0.32333 | -0.0214 | 0.11519 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.2465 | 0.01659 | 0.10568 | -0.3655 | -0.3521 | 0.00968 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.16617 | 0.33837 | -0.2849 | -0.046 | -0.2973 | 0.24585 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.03324 | -0.0133 | -0.0615 | 0.12689 | 0.04979 | -0.001 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.04454 | 0.08697 | 0.27031 | 0.02133 | 0.04598 | -0.2329 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.2045 | 0.05315 | -0.2881 | 0.17749 | 0.05195 | -0.0026 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.046 | 0.09332 | -0.0876 | 0.1184 | 0.19621 | 0.1183 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.432 | 0.13571 | -0.113 | 0.17215 | 0.01917 | 0.05044 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.0484 | -0.0365 | 0.04372 | 0.13451 | 0.04975 | -0.0197 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.08678 | 0.37374 | 0.01626 | 0.103 | 0.019 | 0.05117 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.02999 | 0.05604 | -0.1649 | -0.028 | 0.12594 | 0.103 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.02002 | -0.1481 | -0.1944 | 0.09352 | 0.00926 | 0.11381 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.1727 | -0.0483 | 0.2335 | -0.1161 | -0.0874 | 0.06002 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.19726 | 0.02498 | -0.0312 | 0.18332 | -0.0548 | -0.0732 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.00207 | -0.1106 | 0.18488 | -0.0717 | 0.07699 | 0.11019 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.10792 | 0.00302 | 0.04879 | -0.0084 | -0.2589 | 0.03219 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.23899 | -0.0492 | -0.0236 | 0.1183 | 0.20592 | 0.1194 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.3405 | 0.0268 | 0.393 | -0.0805 | -0.1467 | -0.1738 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.3097 | 0.26271 | -0.137 | 0.01677 | -0.266 | -0.094 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.0274 | 0.02059 | 0.19845 | 0.10062 | 0.17456 | -0.0759 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.2375 | -0.6533 | 0.24419 | -0.0206 | -0.1346 | 0.00716 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.18021 | -0.156 | 0.22465 | -0.0888 | 0.12955 | 0.11253 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.3267 | 0.11546 | -0.0818 | 0.0943 | 0.02337 | 0.10084 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.0652 | -0.1179 | -0.1422 | 0.17143 | 0.11165 | -0.0604 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.11148 | 0.00432 | -0.0707 | -0.0017 | -0.0407 | 0.02047 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.3714 | -0.2546 | -0.0939 | 0.16392 | 0.06805 | 0.09059 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.096 | -0.0243 | 0.30465 | 0.16925 | -0.0355 | 0.0249 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.0143 | -0.0738 | -0.1356 | 0.00371 | 0.12441 | 0.085 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.0298 | -0.1117 | -0.1828 | 0.22396 | 0.00595 | 0.0717 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.3396 | -0.7068 | -0.2048 | 0.13615 | 0.01817 | 0.10753 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.05681 | 0.46601 | -0.0148 | 0.07734 | 0.01795 | -0.0147 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 | 0.15947 | -0.1745 | -0.1905 | -0.1442 | 0.0939 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.5251 | 0.26576 | -0.1393 | -0.0592 | 0.02158 | 0.08939 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.55514 | -0.7534 | -0.2962 | -0.1681 | 0.33885 | -0.1251 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.4185 | 0.19652 | -0.111 | 0.16053 | 0.00337 | -0.0614 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.22991 | 0.01815 | -0.0699 | 0.10827 | -0.0335 | 0.02517 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.06381 | -0.0232 | 0.041 | -0.0442 | -0.1894 | -0.0224 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.4964 | 0.12258 | 0.11779 | 0.32366 | 0.09255 | -0.0121 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.1124 | 0.2255 | 0.38338 | -0.2175 | -0.1109 | 0.06076 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.3331 | 0.1805 | -0.308 | 0.02607 | 0.03676 | -0.1108 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.054 | -0.0439 | 0.07623 | 0.14068 | -0.0486 | 0.16976 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.10501 | 0.06596 | -0.2602 | 0.0165 | -0.0574 | 0.04764 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.19588 | -0.1193 | 0.24184 | 0.09257 | -0.0382 | 0.1975 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.31848 | -0.0371 | -0.2462 | 0.00466 | -0.0194 | -0.0016 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.0291 | -0.0615 | 0.34582 | 0.23521 | 0.00584 | -0.0657 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.00435 | -0.0958 | 0.25759 | -0.0452 | 0.14724 | -0.0794 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.407 | 0.30086 | -0.4387 | 0.00074 | -0.0289 | 0.18007 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.4485 | 0.09305 | -0.0214 | 0.2387 | 0.00506 | 0.03929 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4E-05 | 0.03077 | 0.20145 | 0.31122 | -0.0331 | 0.17877 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.22661 | -0.0053 | -0.03 | 0.07077 | -0.12 | -0.1859 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.05056 | 0.01889 | 0.20309 | 0.09507 | -0.0566 | 0.16704 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.14657 | 0.02527 | 0.40505 | 0.36004 | 0.00626 | 0.15179 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.80974 | -0.1262 | 0.04039 | 0.00603 | 0.11497 | -0.146 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14 | 0.11926 | 0.01305 | -0.083 | -0.0094 | 0.05813 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.0348 | -0.0066 | 0.19462 | 0.31315 | -0.0524 | 0.03431 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.01873 | 0.00108 | -0.0932 | 0.12348 | -0.1179 | 0.03835 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.15039 | 0.27929 | -0.1926 | -0.413 | 0.14339 | 0.14336 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.4131 | 0.18879 | -0.2344 | 0.08656 | 0.12136 | -0.0904 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.23372 | -0.0124 | 0.20339 | 0.0392 | -0.0976 | -0.024 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.04914 | 0.19679 | -0.1033 | 0.06512 | -0.0786 | 0.01133 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.16407 | -0.0312 | -0.0739 | -0.0514 | -0.257 | -0.201 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.24477 | 0.1822 | -0.2571 | -0.0858 | 0.00383 | -0.1693 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.22877 | -0.1288 | 0.13157 | -0.0425 | -0.0489 | 0.03078 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.40488 | 2E-05 | -0.442 | -0.1953 | -0.0199 | 0.02274 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.22541 | -0.1229 | 0.29052 | -0.09 | 0.1076 | -0.0425 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.1559 | 0.05741 | -0.1247 | 0.02238 | 0.00186 | -0.0934 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.118 | -0.0972 | -0.5592 | 0.07689 | 0.04147 | -0.062 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.2665 | -0.3425 | 0.2802 | -0.1693 | 0.17647 | -0.0103 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.104 | 0.26264 | -0.2354 | 0.22795 | 0.017 | -0.0483 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 | -0.535 | 0.05176 | -0.1408 | 0.24745 | 0.12819 |
| - | - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.32437 | -0.1438 | -0.093 | 0.18783 | -0.0205 | 0.28991 |

1. Reduksi menggunakan PCA dengan 11 variabel fitur baru.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| no | *playlist* | pca1 | pca2 | pca3 | pca4 | pca5 | pca6 | pca7 | pca8 | pca9 | pca10 | pca11 |
| 1 | 2 | -0.0091 | -0.3742 | 0.08664 | -0.0268 | -0.1485 | 0.06512 | 0.32701 | -0.0248 | 0.00552 | 0.0637 | -0.1457 |
| 2 | 1 | -1.3658 | -0.0787 | -0.1771 | 0.11925 | 0.15139 | -0.0909 | 0.2377 | 0.09373 | -0.0854 | 0.17142 | 0.02071 |
| 3 | 4 | 0.29293 | -0.4173 | 0.46242 | -0.1193 | 0.08156 | -0.0495 | -0.0394 | -0.0544 | -0.2193 | -0.0462 | 0.06316 |
| 4 | 1 | -1.3266 | -0.1404 | -0.2857 | -0.1449 | 0.12882 | -0.1145 | 0.20513 | -0.0112 | -0.0489 | 0.00561 | 0.03596 |
| 5 | 6 | 0.29202 | -0.5083 | -0.403 | -0.062 | -0.0349 | 0.01449 | -0.0284 | -0.1438 | 0.20002 | 0.22582 | 0.28494 |
| 6 | 4 | 0.58009 | 0.59651 | 0.61325 | -0.0244 | 0.13211 | -0.0549 | 0.0289 | 0.13353 | -0.1019 | 0.14967 | -0.0276 |
| 7 | 5 | 0.29897 | -0.4633 | -0.5835 | 0.37634 | -0.2581 | -0.4956 | 0.13563 | -0.0115 | -0.2616 | -0.1396 | 0.05232 |
| 8 | 5 | 0.71151 | 0.46473 | -0.1306 | -0.0117 | -0.1338 | -0.7184 | -0.184 | -0.0156 | -0.0183 | -0.1347 | 0.02821 |
| 9 | 3 | 0.3556 | -0.4967 | -0.1949 | -0.1727 | 0.03254 | 0.03703 | 0.25735 | -0.0238 | 0.19603 | -0.1432 | -0.049 |
| 10 | 5 | 0.21344 | -0.3884 | 0.12663 | 0.37395 | -0.2193 | -0.5723 | -0.1602 | -0.1337 | -0.0537 | -0.0685 | 0.04717 |
| 11 | 13 | 0.40506 | -0.5186 | 0.06769 | -0.3912 | 0.12329 | 0.00601 | 0.01869 | 0.15042 | 0.00265 | 0.05228 | -0.0753 |
| 12 | 12 | 0.41782 | -0.5281 | -0.1985 | -0.2788 | 0.11709 | 0.02547 | 0.03878 | 0.24211 | -0.1473 | 0.06039 | 0.03729 |
| 13 | 8 | 0.06721 | -0.3101 | -0.4302 | 0.71907 | 0.61397 | 0.19328 | -0.0241 | 0.00837 | 0.0274 | 0.00227 | 0.0036 |
| 14 | 0 | -0.3793 | 0.6998 | -0.059 | -0.1029 | -0.374 | 0.24264 | -0.0495 | -0.122 | -0.0195 | -0.1481 | -0.0374 |
| 15 | 11 | 0.6933 | 0.57028 | 0.567 | 0.03086 | 0.08442 | -0.2526 | -0.0574 | 0.08785 | 0.04424 | -0.0333 | 0.03878 |
| 16 | 5 | -0.2461 | -0.4345 | -0.2766 | -0.1784 | -0.5235 | -0.6602 | -0.2713 | 0.01492 | 0.16259 | 0.01595 | -0.0616 |
| 17 | 6 | 0.55632 | 0.601 | -0.0521 | 0.44524 | -0.0695 | 0.05176 | 0.15986 | -0.1353 | 0.2129 | 0.01113 | 0.20034 |
| 18 | 0 | -0.155 | -0.2606 | 0.35171 | 0.37074 | -0.236 | -0.0768 | 0.26122 | 0.08589 | -0.1485 | 0.14805 | -0.0266 |
| 19 | 0 | -0.6014 | 0.70943 | -0.0843 | -0.2735 | -0.6474 | 0.01464 | 0.08828 | 0.01728 | 0.06802 | 0.04276 | -0.0555 |
| 20 | 2 | -0.1276 | -0.4503 | 0.07083 | -0.5797 | -0.1388 | -0.0235 | -0.1094 | 0.16811 | -0.0266 | 0.15636 | -0.0028 |
| 21 | 1 | -1.2845 | -0.0573 | 0.25215 | 0.09772 | 0.26724 | -0.1411 | 0.13439 | -0.0375 | -0.0209 | 0.13199 | 0.02444 |
| 22 | 4 | 0.05549 | -0.4698 | -0.053 | -0.5444 | 0.67386 | -0.08 | -0.1587 | -0.1066 | -0.0151 | 0.01428 | -0.0778 |
| 23 | 10 | 0.53527 | 0.52163 | -0.1443 | -0.0932 | 0.01152 | 0.07098 | 0.18941 | 0.08956 | -0.0093 | -0.0593 | -0.0651 |
| 24 | 10 | 0.16187 | -0.5085 | 0.00726 | -0.5456 | -0.0097 | -0.0376 | -0.168 | 0.11341 | -0.0319 | -0.018 | 0.06162 |
| 25 | 5 | 0.3328 | -0.4264 | -0.3242 | 0.50143 | -0.2376 | -0.5342 | 0.17538 | -0.0279 | 0.03791 | -0.1151 | -0.0916 |
| 26 | 2 | 0.54624 | 0.51219 | -0.3135 | 0.11091 | -0.1658 | -0.3223 | 0.13974 | -0.0418 | 0.09719 | -0.0006 | -0.1656 |
| 27 | 11 | 0.18314 | -0.4863 | 0.02138 | -0.3027 | -0.0423 | -0.0248 | -0.1836 | -0.3593 | 0.06465 | 0.02257 | 0.1836 |
| 28 | 0 | -0.1791 | -0.4115 | -0.0765 | -0.3472 | -0.1996 | 0.14409 | 0.14834 | -0.2077 | -0.2032 | -0.0632 | -0.0018 |
| 29 | 7 | -0.9654 | 0.80724 | -0.2559 | -0.181 | 0.23135 | 0.01893 | -0.1237 | -0.1841 | 0.09792 | -0.001 | -0.0854 |
| 30 | 2 | -0.0045 | -0.4109 | -0.0571 | -0.0178 | -0.088 | 0.39082 | -0.3415 | -0.1676 | -0.0693 | -0.0613 | -0.0783 |
| 31 | 12 | -0.0097 | -0.3891 | -0.3336 | -0.0841 | 0.6756 | -0.0823 | 0.26149 | 0.32333 | -0.0214 | 0.11519 | 0.00312 |
| 32 | 0 | -0.2065 | 0.67226 | -0.2658 | 0.08937 | -0.2465 | 0.01659 | 0.10568 | -0.3655 | -0.3521 | 0.00968 | -0.0214 |
| 33 | 8 | 0.37701 | -0.4322 | -0.0042 | 0.27432 | 0.16617 | 0.33837 | -0.2849 | -0.046 | -0.2973 | 0.24585 | 0.02738 |
| 34 | 13 | 0.27642 | -0.4933 | 0.00875 | -0.3232 | 0.03324 | -0.0133 | -0.0615 | 0.12689 | 0.04979 | -0.001 | 0.01871 |
| 35 | 3 | 0.34585 | -0.508 | -0.2148 | -0.2831 | 0.04454 | 0.08697 | 0.27031 | 0.02133 | 0.04598 | -0.2329 | -0.0086 |
| 36 | 2 | -0.2774 | -0.3393 | 0.34884 | -0.2182 | -0.2045 | 0.05315 | -0.2881 | 0.17749 | 0.05195 | -0.0026 | 0.14672 |
| 37 | 6 | 0.07976 | -0.3512 | 0.36241 | 0.09836 | -0.046 | 0.09332 | -0.0876 | 0.1184 | 0.19621 | 0.1183 | 0.3337 |
| 38 | 9 | -0.6192 | -0.3246 | -0.1383 | -0.2507 | -0.432 | 0.13571 | -0.113 | 0.17215 | 0.01917 | 0.05044 | 0.00779 |
| 39 | 12 | 0.61952 | 0.43818 | -0.63 | -0.1251 | -0.0484 | -0.0365 | 0.04372 | 0.13451 | 0.04975 | -0.0197 | -0.0223 |
| 40 | 8 | 0.32183 | -0.4778 | -0.7427 | 0.29903 | 0.08678 | 0.37374 | 0.01626 | 0.103 | 0.019 | 0.05117 | 0.07122 |
| 41 | 6 | 0.53986 | 0.47038 | -0.081 | -0.3589 | 0.02999 | 0.05604 | -0.1649 | -0.028 | 0.12594 | 0.103 | 0.19767 |
| 42 | 13 | 0.57015 | 0.49109 | 0.1389 | -0.2977 | 0.02002 | -0.1481 | -0.1944 | 0.09352 | 0.00926 | 0.11381 | -0.0744 |
| 43 | 1 | -1.3899 | -0.0961 | -0.1398 | -0.0395 | 0.1727 | -0.0483 | 0.2335 | -0.1161 | -0.0874 | 0.06002 | 0.02276 |
| 44 | 4 | 0.50801 | -0.5529 | -0.0078 | -0.4582 | 0.19726 | 0.02498 | -0.0312 | 0.18332 | -0.0548 | -0.0732 | 0.08637 |
| 45 | 11 | 0.52067 | 0.53077 | 0.27069 | -0.3072 | 0.00207 | -0.1106 | 0.18488 | -0.0717 | 0.07699 | 0.11019 | 0.18296 |
| 46 | 4 | 0.28339 | -0.3786 | 0.56927 | 0.0174 | 0.10792 | 0.00302 | 0.04879 | -0.0084 | -0.2589 | 0.03219 | 0.02125 |
| 47 | 1 | -0.9152 | 0.85401 | -0.1261 | 0.0588 | 0.23899 | -0.0492 | -0.0236 | 0.1183 | 0.20592 | 0.1194 | -0.1039 |
| 48 | 0 | -0.0682 | 0.56654 | -0.3105 | -0.4993 | -0.3405 | 0.0268 | 0.393 | -0.0805 | -0.1467 | -0.1738 | -0.0024 |
| 49 | 0 | -0.1074 | 0.5659 | -0.5394 | -0.2883 | -0.3097 | 0.26271 | -0.137 | 0.01677 | -0.266 | -0.094 | 0.07416 |
| 50 | 3 | 0.55495 | 0.53179 | -0.0761 | -0.0296 | -0.0274 | 0.02059 | 0.19845 | 0.10062 | 0.17456 | -0.0759 | -0.1464 |
| 51 | 5 | 0.2732 | -0.4157 | -0.1466 | 0.36082 | -0.2375 | -0.6533 | 0.24419 | -0.0206 | -0.1346 | 0.00716 | -0.0443 |
| 52 | 7 | -1.3544 | -0.1587 | -0.2047 | -0.3359 | 0.18021 | -0.156 | 0.22465 | -0.0888 | 0.12955 | 0.11253 | -0.071 |
| 53 | 2 | -0.4262 | -0.2367 | 0.45196 | 0.17739 | -0.3267 | 0.11546 | -0.0818 | 0.0943 | 0.02337 | 0.10084 | -0.0465 |
| 54 | 3 | 0.18595 | -0.4945 | -0.0807 | -0.3447 | -0.0652 | -0.1179 | -0.1422 | 0.17143 | 0.11165 | -0.0604 | -0.0279 |
| 55 | 13 | 0.56233 | 0.416 | -0.0895 | -0.7241 | 0.11148 | 0.00432 | -0.0707 | -0.0017 | -0.0407 | 0.02047 | -0.027 |
| 56 | 0 | -0.5163 | 0.79405 | 0.2502 | 0.15413 | -0.3714 | -0.2546 | -0.0939 | 0.16392 | 0.06805 | 0.09059 | -0.0153 |
| 57 | 13 | 0.07233 | -0.3545 | 0.26107 | 0.05579 | -0.096 | -0.0243 | 0.30465 | 0.16925 | -0.0355 | 0.0249 | -0.0388 |
| 58 | 6 | 0.32191 | -0.4189 | 0.07648 | 0.23684 | -0.0143 | -0.0738 | -0.1356 | 0.00371 | 0.12441 | 0.085 | 0.20672 |
| 59 | 13 | 0.12885 | -0.3711 | 0.50966 | -0.0181 | -0.0298 | -0.1117 | -0.1828 | 0.22396 | 0.00595 | 0.0717 | -0.0051 |
| 60 | 5 | 0.01771 | -0.3843 | 0.10465 | 0.18391 | -0.3396 | -0.7068 | -0.2048 | 0.13615 | 0.01817 | 0.10753 | -0.0171 |
| 61 | 8 | 0.59039 | 0.54937 | -0.4756 | 0.41289 | 0.05681 | 0.46601 | -0.0148 | 0.07734 | 0.01795 | -0.0147 | -0.0242 |
| 62 | 4 | 0.58563 | 0.58581 | 0.51909 | 0.05649 | 0.14186 | 0.15947 | -0.1745 | -0.1905 | -0.1442 | 0.0939 | -0.0854 |
| 63 | 9 | -0.7256 | -0.3249 | -0.2463 | -0.2658 | -0.5251 | 0.26576 | -0.1393 | -0.0592 | 0.02158 | 0.08939 | -0.0637 |
| 64 | 3 | 0.30912 | 0.48059 | -0.3386 | -0.4707 | 0.55514 | -0.7534 | -0.2962 | -0.1681 | 0.33885 | -0.1251 | 0.11671 |
| 65 | 9 | -0.5266 | -0.2257 | 0.23283 | 0.24966 | -0.4185 | 0.19652 | -0.111 | 0.16053 | 0.00337 | -0.0614 | 0.01826 |
| 66 | 4 | 0.3601 | -0.5413 | -0.0634 | -0.5202 | 0.22991 | 0.01815 | -0.0699 | 0.10827 | -0.0335 | 0.02517 | -0.0064 |
| 67 | 4 | 0.24993 | -0.4904 | 0.24073 | -0.5267 | 0.06381 | -0.0232 | 0.041 | -0.0442 | -0.1894 | -0.0224 | 0.01952 |
| 68 | 9 | -0.6475 | -0.2244 | 0.09 | 0.13646 | -0.4964 | 0.12258 | 0.11779 | 0.32366 | 0.09255 | -0.0121 | 0.00841 |
| 69 | 2 | 0.00363 | -0.3416 | 0.12578 | 0.16459 | -0.1124 | 0.2255 | 0.38338 | -0.2175 | -0.1109 | 0.06076 | -0.1629 |
| 70 | 9 | -0.4402 | -0.3552 | 0.01406 | -0.2767 | -0.3331 | 0.1805 | -0.308 | 0.02607 | 0.03676 | -0.1108 | 0.00576 |
| 71 | 11 | 0.18066 | -0.3436 | 0.75268 | -0.0136 | 0.054 | -0.0439 | 0.07623 | 0.14068 | -0.0486 | 0.16976 | -0.0276 |
| 72 | 11 | 0.65118 | 0.3944 | -0.2439 | -0.5844 | 0.10501 | 0.06596 | -0.2602 | 0.0165 | -0.0574 | 0.04764 | 0.02318 |
| 73 | 7 | -1.3577 | -0.1472 | -0.3721 | -0.1908 | 0.19588 | -0.1193 | 0.24184 | 0.09257 | -0.0382 | 0.1975 | -0.0054 |
| 74 | 7 | -0.8608 | 0.80326 | -0.1491 | -0.157 | 0.31848 | -0.0371 | -0.2462 | 0.00466 | -0.0194 | -0.0016 | 0.02294 |
| 75 | 12 | 0.46947 | 0.48715 | -0.6646 | -0.1092 | -0.0291 | -0.0615 | 0.34582 | 0.23521 | 0.00584 | -0.0657 | 0.08233 |
| 76 | 3 | 0.23401 | -0.4187 | -0.2087 | 0.21926 | 0.00435 | -0.0958 | 0.25759 | -0.0452 | 0.14724 | -0.0794 | -0.191 |
| 77 | 2 | -0.5713 | -0.365 | -0.2871 | -0.2304 | -0.407 | 0.30086 | -0.4387 | 0.00074 | -0.0289 | 0.18007 | -0.0376 |
| 78 | 9 | -0.5974 | -0.2438 | 0.23343 | 0.02045 | -0.4485 | 0.09305 | -0.0214 | 0.2387 | 0.00506 | 0.03929 | 0.03564 |
| 79 | 12 | 0.53978 | 0.5439 | -0.2507 | 0.12956 | -4E-05 | 0.03077 | 0.20145 | 0.31122 | -0.0331 | 0.17877 | -0.0222 |
| 80 | 1 | -0.8903 | 0.8734 | -0.1096 | 0.15616 | 0.22661 | -0.0053 | -0.03 | 0.07077 | -0.12 | -0.1859 | 0.09963 |
| 81 | 2 | 0.29067 | -0.3725 | 0.36626 | 0.1919 | 0.05056 | 0.01889 | 0.20309 | 0.09507 | -0.0566 | 0.16704 | -0.1132 |
| 82 | 12 | 0.37103 | 0.66894 | -0.2725 | 0.60819 | 0.14657 | 0.02527 | 0.40505 | 0.36004 | 0.00626 | 0.15179 | 0.13063 |
| 83 | 3 | 0.19678 | 0.67594 | 0.1521 | 0.08227 | 0.80974 | -0.1262 | 0.04039 | 0.00603 | 0.11497 | -0.146 | -0.1016 |
| 84 | 10 | -0.1118 | -0.3192 | 0.48405 | -0.0002 | -0.14 | 0.11926 | 0.01305 | -0.083 | -0.0094 | 0.05813 | -0.0857 |
| 85 | 2 | 0.12251 | -0.3481 | 0.30009 | 0.13513 | -0.0348 | -0.0066 | 0.19462 | 0.31315 | -0.0524 | 0.03431 | 0.00697 |
| 86 | 13 | 0.16322 | -0.473 | 0.12468 | -0.4162 | 0.01873 | 0.00108 | -0.0932 | 0.12348 | -0.1179 | 0.03835 | 0.01663 |
| 87 | 6 | 0.36675 | -0.4069 | 0.49634 | 0.0793 | 0.15039 | 0.27929 | -0.1926 | -0.413 | 0.14339 | 0.14336 | 0.31126 |
| 88 | 9 | -0.5323 | -0.35 | -0.0753 | -0.314 | -0.4131 | 0.18879 | -0.2344 | 0.08656 | 0.12136 | -0.0904 | -0.0257 |
| 89 | 0 | -0.2051 | 0.62893 | -0.4943 | -0.2516 | 0.23372 | -0.0124 | 0.20339 | 0.0392 | -0.0976 | -0.024 | -0.0302 |
| 90 | 8 | 0.45093 | -0.4135 | -0.3373 | 0.6831 | 0.04914 | 0.19679 | -0.1033 | 0.06512 | -0.0786 | 0.01133 | 0.09028 |
| 91 | 11 | 0.46221 | -0.4492 | 0.39011 | -0.0744 | 0.16407 | -0.0312 | -0.0739 | -0.0514 | -0.257 | -0.201 | 0.07101 |
| 92 | 7 | -0.8762 | 0.8638 | -0.3148 | 0.33352 | 0.24477 | 0.1822 | -0.2571 | -0.0858 | 0.00383 | -0.1693 | -0.0152 |
| 93 | 1 | -1.2488 | -0.0982 | 0.14988 | -0.054 | 0.22877 | -0.1288 | 0.13157 | -0.0425 | -0.0489 | 0.03078 | 0.04534 |
| 94 | 7 | -0.9279 | -0.1864 | 0.17539 | -0.0786 | 0.40488 | 2E-05 | -0.442 | -0.1953 | -0.0199 | 0.02274 | 0.07719 |
| 95 | 11 | -0.4082 | -0.3456 | -0.0578 | -0.3885 | 0.22541 | -0.1229 | 0.29052 | -0.09 | 0.1076 | -0.0425 | -0.0801 |
| 96 | 13 | -0.1277 | -0.4331 | 0.09377 | -0.491 | -0.1559 | 0.05741 | -0.1247 | 0.02238 | 0.00186 | -0.0934 | -0.0127 |
| 97 | 11 | 0.26866 | 0.64093 | 0.51422 | 0.12077 | -0.118 | -0.0972 | -0.5592 | 0.07689 | 0.04147 | -0.062 | 0.03556 |
| 98 | 10 | 0.13333 | -0.4752 | -0.3415 | -0.0896 | -0.2665 | -0.3425 | 0.2802 | -0.1693 | 0.17647 | -0.0103 | -0.2112 |
| 99 | 2 | 0.01009 | -0.2853 | 0.24447 | 0.55205 | -0.104 | 0.26264 | -0.2354 | 0.22795 | 0.017 | -0.0483 | -0.009 |
| 100 | 6 | 0.62453 | 0.42892 | -0.6022 | -0.0802 | -0.2063 | -0.535 | 0.05176 | -0.1408 | 0.24745 | 0.12819 | 0.27719 |
| - | - | - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - | - | - |
| - | - | - | - | - | - | - | - | - | - | - | - | - |
| 700 | 1 | -1.1201 | -0.0958 | 0.22272 | 0.12806 | 0.32437 | -0.1438 | -0.093 | 0.18783 | -0.0205 | 0.28991 | 0.03483 |

**Lampiran C.2.** *Source Code* PCAKNN Pengujian Kedua

|  |  |
| --- | --- |
| No | *Source Code* |
| 1 | public *function* KlasifikasiPCAKNN($pca,$nilaiK){ |
| 2 | $hasilShuffle['indexPca'] = $pca; |
| 3 | $hasilShuffle['data'] = $this->Data->ViewDataPca("data\_pca".$pca); |
| 4 | for($f=0; $f<10; $f++){ |
| 5 | $data0 = array(); |
| 6 | $data1 = array(); |
| 7 | $data2 = array(); |
| 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; |

|  |  |
| --- | --- |
| No | *Source Code* |
| 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, $hasilShuffle['data'][$j]); |
| 54 | array\_push($data0, $hasilShuffle['data'][$j]); |
| 55 | }else{ |
| 56 | $dataIndex0++; |
| 57 | } |
| 58 | } |
| 59 | if($hasilShuffle['data'][$j]->playlist == 1){ |
| 60 | if(count($data1)<5&& $dataIndex1 == $f\*5){ |
| 61 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 62 | array\_push($data1, $hasilShuffle['data'][$j]); |
| 63 | }else{ |
| 64 | $dataIndex1++; |
| 65 | } |

|  |  |
| --- | --- |
| No | *Source Code* |
| 66 | } |
| 67 | if($hasilShuffle['data'][$j]->*playlist* == 2){ |
| 68 | if(count($data2)<5&& $dataIndex2 == $f\*5){ |
| 69 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 70 | array\_push($data2, $hasilShuffle['data'][$j]); |
| 71 | }else{ |
| 72 | $dataIndex2++; |
| 73 | } |
| 74 | } |
| 75 | if($hasilShuffle['data'][$j]->*playlist* == 3){ |
| 76 | if(count($data3)<5&& $dataIndex3 == $f\*5){ |
| 77 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 78 | array\_push($data3, $hasilShuffle['data'][$j]); |
| 79 | }else{ |
| 80 | $dataIndex3++; |
| 81 | } |
| 82 | } |
| 83 | if($hasilShuffle['data'][$j]->*playlist* == 4){ |
| 84 | if(count($data4)<5&& $dataIndex4 == $f\*5){ |
| 85 | array\_push($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]); |

|  |  |
| --- | --- |
| No | *Source Code* |
| 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{ |
| 120 | $dataIndex8++; |
| 121 | } |
| 122 | } |
| 123 | if($hasilShuffle['data'][$j]->*playlist* == 9){ |
| 124 | if(count($data9)<5&& $dataIndex9 == $f\*5){ |
| 125 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 126 | array\_push($data9, $hasilShuffle['data'][$j]); |
| 127 | }else{ |
| 128 | $dataIndex9++; |
| 129 | } |
| 130 | } |
| 131 | if($hasilShuffle['data'][$j]->playlist == 10){ |
| 132 | if(count($data10)<5&& $dataIndex10 == $f\*5){ |
| 133 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 134 | array\_push($data10, $hasilShuffle['data'][$j]); |
| 135 | }else{ |
| 136 | $dataIndex10++; |
| 137 | } |

|  |  |
| --- | --- |
| No | *Source Code* |
| 138 | } |
| 139 | if($hasilShuffle['data'][$j]->*playlist* == 11){ |
| 140 | if(count($data11)<5&& $dataIndex11 == $f\*5){ |
| 141 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 142 | array\_push($data11, $hasilShuffle['data'][$j]); |
| 143 | }else{ |
| 144 | $dataIndex11++; |
| 145 | } |
| 146 | } |
| 147 | if($hasilShuffle['data'][$j]->*playlist* == 12){ |
| 148 | if(count($data12)<5&& $dataIndex12 == $f\*5){ |
| 149 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 150 | array\_push($data12, $hasilShuffle['data'][$j]); |
| 151 | }else{ |
| 152 | $dataIndex12++; |
| 153 | } |
| 154 | } |
| 155 | if($hasilShuffle['data'][$j]->*playlist* == 13){ |
| 156 | if(count($data13)<5&& $dataIndex13 == $f\*5){ |
| 157 | array\_push($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); |
| 166 | $jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 167 | for($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++){ |

|  |  |
| --- | --- |
| No | *Source Code* |
| 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 | $totalPow = 0; |
| 182 | for($p=0;$p<$pca;$p++){ |
| 183 | $fitur = "pca".($p+1); |
| 184 | $totalPow += pow(abs($dataTesting[$j]->$fitur - $hasilShuffle['data'][$k]->$fitur), 2); |
| 185 | } |
| 186 | $kalkulasiDistance->distance = sqrt($totalPow); |
| 187 | array\_push($hasilDistance, $kalkulasiDistance); |
| 188 | } |
| 189 | } |
| 190 | usort($hasilDistance, "cmp"); |
| 191 | for($k=0; $k<$nilaiK; $k++){ |
| 192 | $hasilShuffle['hasilDistance'][$f][$j][$k] = $hasilDistance[$k]; |
| 193 | } |
| 194 | $hasilClass = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 195 | for($k=0;$k<$nilaiK;$k++){ |
| 196 | $hasilClass[$hasilShuffle['data'][$hasilDistance[$k]->no-1]->*playlist*] +=1; |
| 197 | } |
| 198 | $max = $hasilClass[0]; |
| 199 | $classHigh = 0; |
| 200 | for($k=1;$k<count($hasilClass);$k++){ |
| 201 | if($max<$hasilClass[$k]){ |
| 202 | $max = $hasilClass[$k]; |
| 203 | $classHigh = $k; |
| 204 | } |
| 205 | } |
| 206 | $hasilShuffle['dataHasilTesting'][$f][$j] = $classHigh; |
| 207 | if($classHigh == $hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist){ |
| 208 | $jumlahBenar[$hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist]++; |
| 209 | }else { |

|  |  |
| --- | --- |
| No | *Source Code* |
| 210 | $jumlahSalah[$hasilShuffle['data'][$dataTesting[$j]->no-1]->*playlist*]++; |
| 211 | } |
| 212 | switch ($hasilShuffle['data'][$dataTesting[$j]->no-1]->*playlist*) { |
| 213 | case 0: |
| 214 | $dataHasil0[$classHigh]++; |
| 215 | break; |
| 216 | case 1: |
| 217 | $dataHasil1[$classHigh]++; |
| 218 | break; |
| 219 | case 2: |
| 220 | $dataHasil2[$classHigh]++; |
| 221 | break; |
| 222 | case 3: |
| 223 | $dataHasil3[$classHigh]++; |
| 224 | break; |
| 225 | case 4: |
| 226 | $dataHasil4[$classHigh]++; |
| 227 | break; |
| 228 | case 5: |
| 229 | $dataHasil5[$classHigh]++; |
| 230 | break; |
| 231 | case 6: |
| 232 | $dataHasil6[$classHigh]++; |
| 233 | break; |
| 234 | case 7: |
| 235 | $dataHasil7[$classHigh]++; |
| 236 | break; |
| 237 | case 8: |
| 238 | $dataHasil8[$classHigh]++; |
| 239 | break; |
| 240 | case 9: |
| 241 | $dataHasil9[$classHigh]++; |
| 242 | break; |
| 243 | case 10: |
| 244 | $dataHasil10[$classHigh]++; |
| 245 | break; |
| 246 | case 11: |
| 247 | $dataHasil11[$classHigh]++; |
| 248 | break; |
| 249 | case 12: |

|  |  |
| --- | --- |
| No | *Source Code* |
| 250 | $dataHasil12[$classHigh]++; |
| 251 | break; |
| 252 | case 13: |
| 253 | $dataHasil13[$classHigh]++; |
| 254 | break; |
| 255 | } |
| 256 | } |
| 257 | $hasilShuffle["DataMatriks"][$f][0] = $dataHasil0; |
| 258 | $hasilShuffle["DataMatriks"][$f][1] = $dataHasil1; |
| 259 | $hasilShuffle["DataMatriks"][$f][2] = $dataHasil2; |
| 260 | $hasilShuffle["DataMatriks"][$f][3] = $dataHasil3; |
| 261 | $hasilShuffle["DataMatriks"][$f][4] = $dataHasil4; |
| 262 | $hasilShuffle["DataMatriks"][$f][5] = $dataHasil5; |
| 263 | $hasilShuffle["DataMatriks"][$f][6] = $dataHasil6; |
| 264 | $hasilShuffle["DataMatriks"][$f][7] = $dataHasil7; |
| 265 | $hasilShuffle["DataMatriks"][$f][8] = $dataHasil8; |
| 266 | $hasilShuffle["DataMatriks"][$f][9] = $dataHasil9; |
| 267 | $hasilShuffle["DataMatriks"][$f][10] = $dataHasil10; |
| 268 | $hasilShuffle["DataMatriks"][$f][11] = $dataHasil11; |
| 269 | $hasilShuffle["DataMatriks"][$f][12] = $dataHasil12; |
| 270 | $hasilShuffle["DataMatriks"][$f][13] = $dataHasil13; |
| 271 | $calculateBenar = 0; |
| 272 | for($j=0; $j<count($jumlahBenar); $j++){ |
| 273 | $calculateBenar +=  $jumlahBenar[$j]; |
| 274 | } |
| 275 | $hasilShuffle["HasilAkurasi"][$f] = $calculateBenar/70\*100; |
| 276 | } |
| 277 | $this->load->view('ViewKlasifikasiPCAKNN', $hasilShuffle); |
| 278 | } |

**Lampiran C.3.** *Source Code* PCAMKNN Pengujian Kedua

|  |  |
| --- | --- |
| No | *Source Code* |
| 1 | *public function KlasifikasiPCAMKNN($pca,$nilaiK){* |
| 2 | *$hasilShuffle['indexPca'] = $pca;* |
| 3 | *$hasilShuffle['data'] = $this->Data->ViewDataPca("data\_pca".$pca);* |
| 4 | *for($g=0; $g<10; $g++){* |
| 5 | *$data0 = array();* |

|  |  |
| --- | --- |
| No | *Source Code* |
| 6 | $data1 = array(); |
| 7 | $data2 = array(); |
| 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(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 37 | $hasilData1 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 38 | $hasilData2 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 39 | $hasilData3 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 40 | $hasilData4 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 41 | $hasilData5 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 42 | $hasilData6 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 43 | $hasilData7 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 44 | $hasilData8 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |

|  |  |
| --- | --- |
| No | *Source Code* |
| 45 | $hasilData9 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 46 | $hasilData10 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 47 | $hasilData11 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 48 | $hasilData12 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 49 | $hasilData13 = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 50 | for($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, $hasilShuffle['data'][$j]); |
| 54 | array\_push($data0, $hasilShuffle['data'][$j]); |
| 55 | }else{ |
| 56 | $dataIndex0++; |
| 57 | } |
| 58 | } |
| 59 | if($hasilShuffle['data'][$j]->*playlist* == 1){ |
| 60 | if(count($data1)<5&& $dataIndex1 == $g\*5){ |
| 61 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 62 | array\_push($data1, $hasilShuffle['data'][$j]); |
| 63 | }else{ |
| 64 | $dataIndex1++; |
| 65 | } |
| 66 | } |
| 67 | if($hasilShuffle['data'][$j]->*playlist* == 2){ |
| 68 | if(count($data2)<5&& $dataIndex2 == $g\*5){ |
| 69 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 70 | array\_push($data2, $hasilShuffle['data'][$j]); |
| 71 | }else{ |
| 72 | $dataIndex2++; |
| 73 | } |
| 74 | } |
| 75 | if($hasilShuffle['data'][$j]->*playlist* == 3){ |
| 76 | if(count($data3)<5&& $dataIndex3 == $g\*5){ |
| 77 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 78 | array\_push($data3, $hasilShuffle['data'][$j]); |
| 79 | }else{ |
| 80 | $dataIndex3++; |
| 81 | } |

|  |  |
| --- | --- |
| No | *Source Code* |
| 82 | } |
| 83 | if($hasilShuffle['data'][$j]->*playlist* == 4){ |
| 84 | if(count($data4)<5&& $dataIndex4 == $g\*5){ |
| 85 | array\_push($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 == $g\*5){ |
| 93 | array\_push($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 == $g\*5){ |
| 101 | array\_push($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 == $g\*5){ |
| 109 | array\_push($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 == $g\*5){ |
| 117 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |

|  |  |
| --- | --- |
| No | *Source Code* |
| 118 | array\_push($data8, $hasilShuffle['data'][$j]); |
| 119 | }else{ |
| 120 | $dataIndex8++; |
| 121 | } |
| 122 | } |
| 123 | if($hasilShuffle['data'][$j]->*playlist* == 9){ |
| 124 | if(count($data9)<5&& $dataIndex9 == $g\*5){ |
| 125 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 126 | array\_push($data9, $hasilShuffle['data'][$j]); |
| 127 | }else{ |
| 128 | $dataIndex9++; |
| 129 | } |
| 130 | } |
| 131 | if($hasilShuffle['data'][$j]->*playlist* == 10){ |
| 132 | if(count($data10)<5&& $dataIndex10 == $g\*5){ |
| 133 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 134 | array\_push($data10, $hasilShuffle['data'][$j]); |
| 135 | }else{ |
| 136 | $dataIndex10++; |
| 137 | } |
| 138 | } |
| 139 | if($hasilShuffle['data'][$j]->*playlist* == 11){ |
| 140 | if(count($data11)<5&& $dataIndex11 == $g\*5){ |
| 141 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 142 | array\_push($data11, $hasilShuffle['data'][$j]); |
| 143 | }else{ |
| 144 | $dataIndex11++; |
| 145 | } |
| 146 | } |
| 147 | if($hasilShuffle['data'][$j]->playlist == 12){ |
| 148 | if(count($data12)<5&& $dataIndex12 == $g\*5){ |
| 149 | array\_push($allDataTesting, $hasilShuffle['data'][$j]); |
| 150 | array\_push($data12, $hasilShuffle['data'][$j]); |
| 151 | }else{ |
| 152 | $dataIndex12++; |
| 153 | } |
| 154 | } |

|  |  |
| --- | --- |
| No | *Source Code* |
| 155 | if($hasilShuffle['data'][$j]->*playlist* == 13){ |
| 156 | if(count($data13)<5&& $dataIndex13 == $g\*5){ |
| 157 | array\_push($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); |
| 166 | $jumlahSalah = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 167 | for($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 | $totalPow = 0; |
| 182 | for($p=0;$p<$pca;$p++){ |
| 183 | $fitur = "pca".($p+1); |
| 184 | $totalPow += pow(abs($dataTesting[$j]->$fitur - $hasilShuffle['data'][$k]->$fitur), 2); |
| 185 | } |
| 186 | $kalkulasiDistance->distance = sqrt($totalPow); |
| 187 | array\_push($hasilDistance, $kalkulasiDistance); |
| 188 | } |
| 189 | } |
| 190 | usort($hasilDistance, "cmp"); |
| 191 | for($k=0; $k<$nilaiK; $k++){ |

|  |  |
| --- | --- |
| No | *Source Code* |
| 192 | $hasilShuffle['hasilDistance'][$g][$j][$k] = $hasilDistance[$k]; |
| 193 | } |
| 194 | $weightResult = array(); |
| 195 | for($i=0;$i<$nilaiK;$i++){ |
| 196 | $dataTest = $hasilShuffle['data'][$hasilDistance[$i]->no-1]; |
| 197 | $hasilShuffle['dataTestingResult'][$g][$j][$i] = $dataTest; |
| 198 | $resultDistance = array(); |
| 199 | for($k=0;$k<count($hasilShuffle['data']);$k++){ |
| 200 | $sama = 0; |
| 201 | for($l=0; $l<count($allDataTesting);$l++){ |
| 202 | if($allDataTesting[$l]->no == $hasilShuffle['data'][$k]->no){ |
| 203 | $sama = 1; |
| 204 | break; |
| 205 | } |
| 206 | } |
| 207 | if($sama == 0 && $dataTest->no != $hasilShuffle['data'][$k]->no){ |
| 208 | $kalDistance = new *MyDistance*(); |
| 209 | $kalDistance->no = $hasilShuffle['data'][$k]->no; |
| 210 | $totalPow = 0; |
| 211 | for($p=0;$p<$pca;$p++){ |
| 212 | $fitur = "pca".($p+1); |
| 213 | $totalPow += pow(abs($dataTest->$fitur - $hasilShuffle['data'][$k]->$fitur), 2); |
| 214 | } |
| 215 | $kalkulasiDistance->distance = sqrt($totalPow); |
| 216 | array\_push($resultDistance, $kalDistance); |
| 217 | } |
| 218 | } |
| 219 | usort($resultDistance, "cmp"); |
| 220 | for($k=0; $k<$nilaiK; $k++){ |
| 221 | $hasilShuffle['resultDistance'][$g][$j][$i][$k] = $resultDistance[$k]; |
| 222 | } |
| 223 | $validasi = 0; |
| 224 | for($k=0; $k<$nilaiK; $k++){ |
| 225 | if($hasilShuffle['data'][$resultDistance[$k]->no-1]->playlist == $dataTest->playlist){ |
| 226 | $validasi++; |

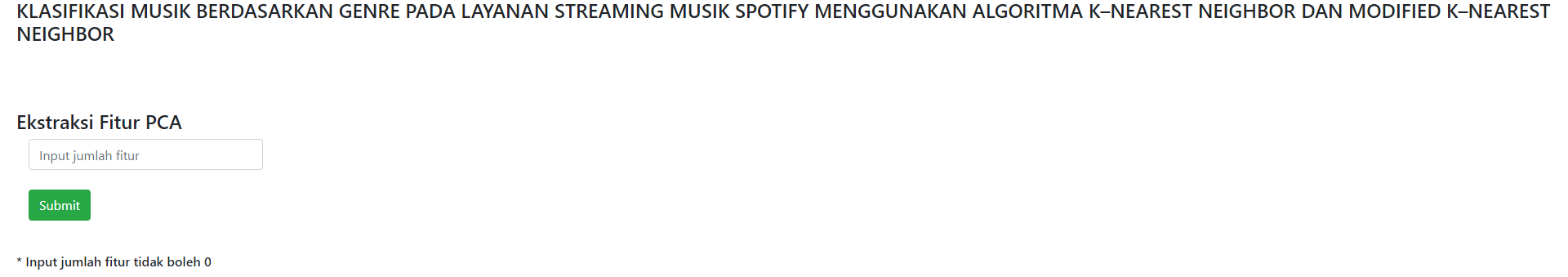
|  |  |
| --- | --- |
| No | *Source Code* |
| 227 | } |
| 228 | } |
| 229 | $validasi = $validasi/$nilaiK; |
| 230 | $hasilShuffle['Validasi'][$g][$j][$i] = $validasi; |
| 231 | $cacheWeight = $validasi \* (1/($hasilDistance[$i]->distance+0.5)); |
| 232 | $hasilShuffle['Weight'][$g][$j][$i] = $cacheWeight; |
| 233 | $classWight = new *MyWeight*(); |
| 234 | $classWight->*playlist* = $dataTest->*playlist*; |
| 235 | $classWight->weight = $cacheWeight; |
| 236 | array\_push($weightResult, $classWight); |
| 237 | } |
| 238 | $hasilClassDistance = array(0,0,0,0,0,0,0,0,0,0,0,0,0,0); |
| 239 | for($k=0;$k<count($weightResult);$k++){ |
| 240 | $hasilClassDistance[$weightResult[$k]->*playlist*]+=$weightResult[$k]->weight; |
| 241 | } |
| 242 | $choseClass = 0; |
| 243 | $cacheMax = 0; |
| 244 | for($k=0;$k<count($hasilClassDistance);$k++){ |
| 245 | if($cacheMax<$hasilClassDistance[$k]){ |
| 246 | $cacheMax = $hasilClassDistance[$k]; |
| 247 | $choseClass = $k; |
| 248 | } |
| 249 | } |
| 250 | $hasilShuffle['dataHasilTesting'][$g][$j] = $choseClass; |
| 251 | if($choseClass == $hasilShuffle['data'][$dataTesting[$j]->no-1]->*playlist*){ |
| 252 | $jumlahBenar[$hasilShuffle['data'][$dataTesting[$j]->no-1]->*playlist*]++; |
| 253 | }else { |
| 254 | $jumlahSalah[$hasilShuffle['data'][$dataTesting[$j]->no-1]->*playlist*]++; |
| 255 | } |
| 256 | switch ($hasilShuffle['data'][$dataTesting[$j]->no-1]->playlist) { |
| 257 | case 0: |
| 258 | $hasilData0[$choseClass]++; |
| 259 | break; |
| 260 | case 1: |
| 261 | $hasilData1[$choseClass]++; |

|  |  |
| --- | --- |
| No | *Source Code* |
| 262 | break; |
| 263 | case 2: |
| 264 | $hasilData2[$choseClass]++; |
| 265 | break; |
| 266 | case 3: |
| 267 | $hasilData3[$choseClass]++; |
| 268 | break; |
| 269 | case 4: |
| 270 | $hasilData4[$choseClass]++; |
| 271 | break; |
| 272 | case 5: |
| 273 | $hasilData5[$choseClass]++; |
| 274 | break; |
| 275 | case 6: |
| 276 | $hasilData6[$choseClass]++; |
| 277 | break; |
| 278 | case 7: |
| 279 | $hasilData7[$choseClass]++; |
| 280 | break; |
| 281 | case 8: |
| 282 | $hasilData8[$choseClass]++; |
| 283 | break; |
| 284 | case 9: |
| 285 | $hasilData9[$choseClass]++; |
| 286 | break; |
| 287 | case 10: |
| 288 | $hasilData10[$choseClass]++; |
| 289 | break; |
| 290 | case 11: |
| 291 | $hasilData11[$choseClass]++; |
| 292 | break; |
| 293 | case 12: |
| 294 | $hasilData12[$choseClass]++; |
| 295 | break; |
| 296 | case 13: |
| 297 | $hasilData13[$choseClass]++; |
| 298 | break; |
| 299 | } |
| 300 | } |

|  |  |
| --- | --- |
| No | *Source Code* |
| 301 | $hasilShuffle["DataMatriks"][$g][0] = $hasilData0; |
| 302 | $hasilShuffle["DataMatriks"][$g][1] = $hasilData1; |
| 303 | $hasilShuffle["DataMatriks"][$g][2] = $hasilData2; |
| 304 | $hasilShuffle["DataMatriks"][$g][3] = $hasilData3; |
| 305 | $hasilShuffle["DataMatriks"][$g][4] = $hasilData4; |
| 306 | $hasilShuffle["DataMatriks"][$g][5] = $hasilData5; |
| 307 | $hasilShuffle["DataMatriks"][$g][6] = $hasilData6; |
| 308 | $hasilShuffle["DataMatriks"][$g][7] = $hasilData7; |
| 309 | $hasilShuffle["DataMatriks"][$g][8] = $hasilData8; |
| 310 | $hasilShuffle["DataMatriks"][$g][9] = $hasilData9; |
| 311 | $hasilShuffle["DataMatriks"][$g][10] = $hasilData10; |
| 312 | $hasilShuffle["DataMatriks"][$g][11] = $hasilData11; |
| 313 | $hasilShuffle["DataMatriks"][$g][12] = $hasilData12; |
| 314 | $hasilShuffle["DataMatriks"][$g][13] = $hasilData13; |
| 315 | $calculateBenar = 0; |
| 316 | for($j=0; $j<count($jumlahBenar); $j++){ |
| 317 | $calculateBenar +=  $jumlahBenar[$j]; |
| 318 | } |
| 319 | $hasilShuffle["HasilAkurasi"][$g] = $calculateBenar/70\*100; |
| 320 | } |
| 321 | $this->load->view('ViewKlasifikasiPCAMKNN', $hasilShuffle); |
| 322 | } |

**Lampiran C.4.** Implementasi Sistem pada PCAKNN dan PCAMKNN Pengujian Kedua

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



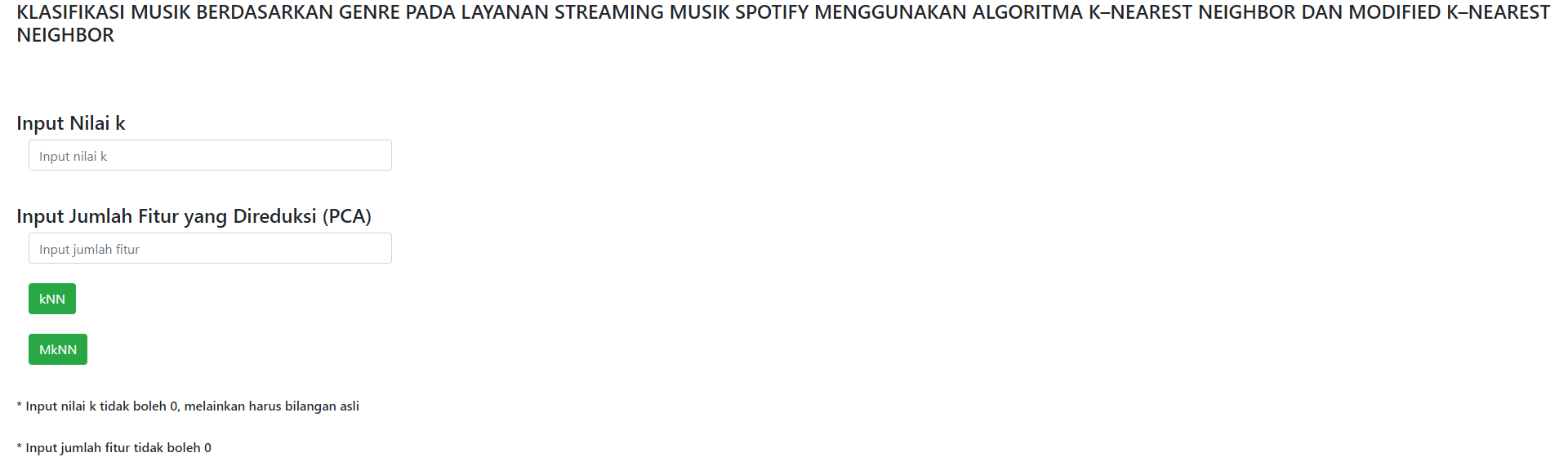
Gambar 1. Input Jumlah Fitur direduksi PCA

Output tampilan hasil jumlah fitur yang telah direduksi. Pada Gambar 2, menampilkan jumlah fitur yang direduksi yaitu 1 fitur.



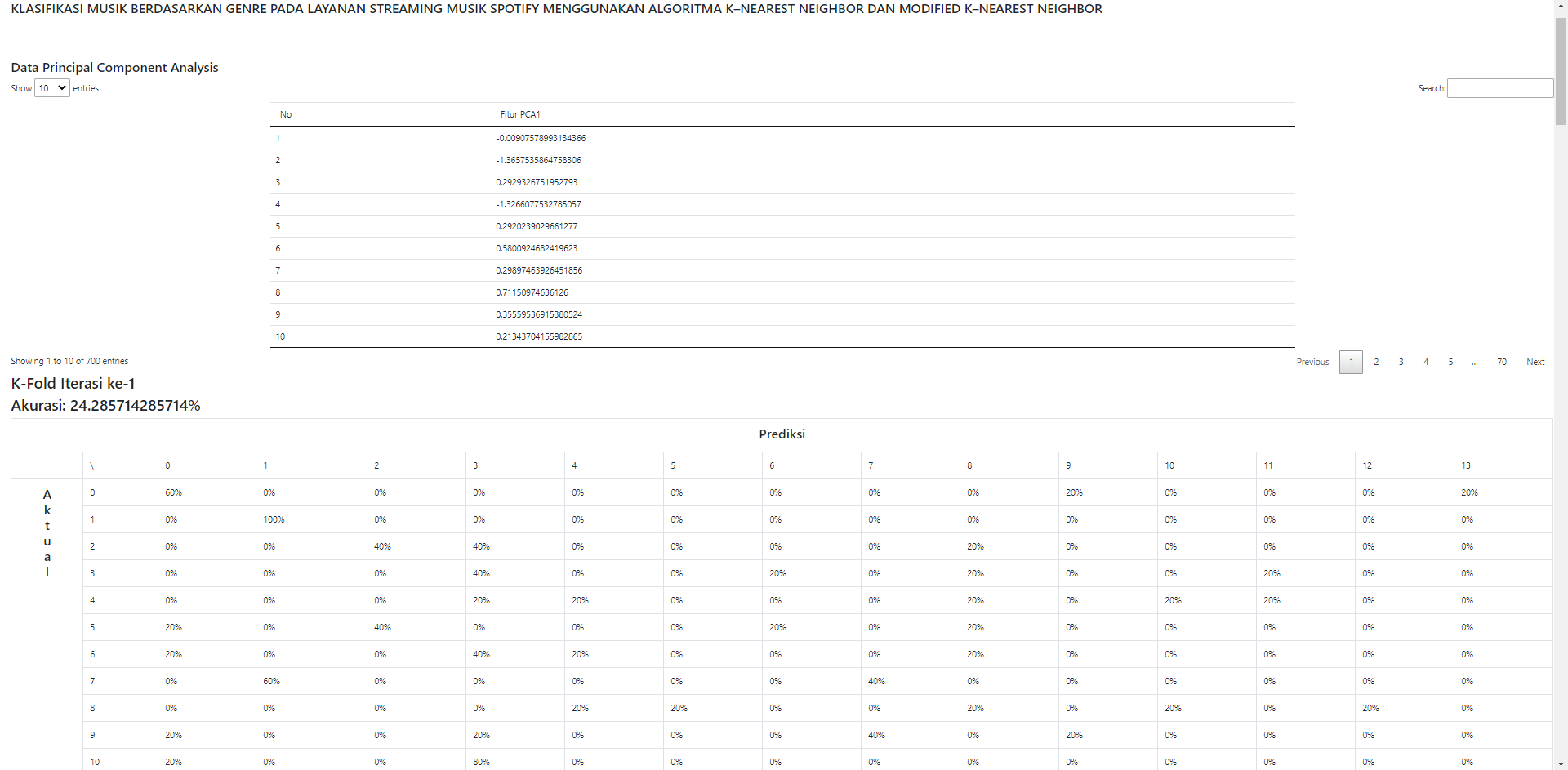
Gambar 2. Jumlah Fitur 1 yang telah direduksi

Pengguna memasukkan nilai *k* dan jumlah fitur direduskiyang nantinya akan digunakan untuk proses KNN dan MKNN. Nilai *k* dibatasi hanya bilangan asli (bilangan bulat positif yang bukan 0) dan input jumlah fitur yang direduksi dengan PCA, tapi nilai inputan tidak boleh 0. Seperti pada Gambar 3.



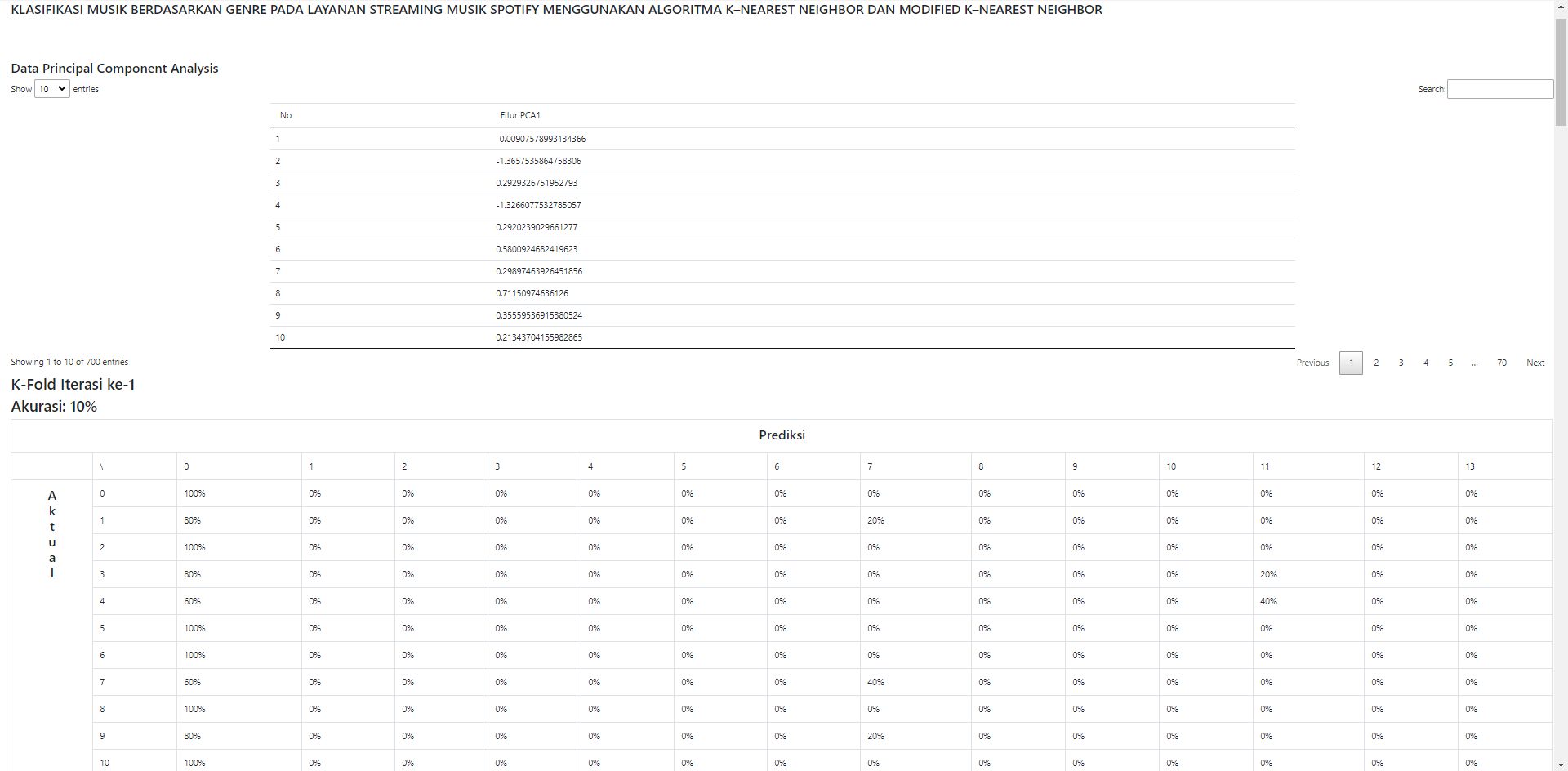
Gambar 3. Input Nilai k dan Jumlah Fitur direduksi PCA

Output dari PCAKNN akan munjul jika nilai *k* telah diinputkan dan jumlah fitur juga diinputkan. Tampilan pada Gambar 4, merupakan hasil dari klasifikasi PCAKNN.



Gambar 4. Output Klasifikasi PCAKNN

Output dari PCAMKNN akan munjul jika nilai *k* telah diinputkan dan jumlah fitur juga diinputkan. Tampilan pada Gambar 5, merupakan hasil dari klasifikasi PCAMKNN.



Gambar 5. Output Klasifikasi PCAMKNN

**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 | | | | | | |
| MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 56.70% | 60.10% | 60.40% | 62% | 61.10% | 61% | 60.40% |
| k=9 | k=10 | k=11 | k=12 | k=13 | k=14 | k=15 |
| 59.60% | 58.60% | 59% | 59% | 59.10% | 58.90% | 58.90% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 58.60% | 56.70% | 57.10% | 57% | 56.10% | 56.40% | 55.70% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 54.90% | 54.90% | 54.10% | 53.40% | 52.90% | 52.70% | 52.40% |
| KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 57.40% | 60.40% | 61.40% | 63% | 61.90% | 63.30% | 62.30% |
| k=9 | k=10 | k=11 | k=12 | k=13 | k=14 | k=15 |
| 62.30% | 61% | 61.90% | 61.40% | 61.60% | 61.10% | 61.70% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 61% | 60.60% | 60.30% | 60.60% | 60.70% | 60.10% | 59.60% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 59.90% | 58.70% | 59% | 58.70% | 58.10% | 57.90% | 57.70% |

**Lampiran D.2.** Disajikan Hasil Perbandingan Akurasi 10*–Fold Cross–Validation* MKNN dengan KNN pada *k* = 5 di Pengujian Pertama

|  |  |  |
| --- | --- | --- |
| 10-Fold Cross-Validation | k=5 MKNN | k=5 KNN |
| 1 | 60% | 62.90% |
| 2 | 60% | 62.90% |
| 3 | 70% | 70% |
| 4 | 64.30% | 65.70% |
| 5 | 58.60% | 55.70% |
| 6 | 58.60% | 60% |
| 7 | 62.90% | 68.60% |
| 8 | 68.60% | 65.70% |
| 9 | 61.40% | 60% |
| 10 | 55.70% | 58.60% |

**Lampiran D.3.** Disajikan Hasil Perbandingan Akurasi 10*–Fold Cross–Validation* MKNN dengan KNN pada *k* = 7 di Pengujian Pertama

|  |  |  |
| --- | --- | --- |
| 10-Fold Cross-Validation | k=7 MKNN | k=7 KNN |
| 1 | 55.70% | 58.60% |
| 2 | 62.90% | 61.40% |
| 3 | 64.30% | 67.10% |
| 4 | 61.40% | 64.30% |
| 5 | 58.60% | 58.60% |
| 6 | 61.40% | 68.60% |
| 7 | 64.30% | 70% |
| 8 | 68.60% | 71.40% |
| 9 | 58.60% | 55.70% |
| 10 | 54.30% | 57.10% |

**Lampiran D.4.** Disajikan Hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold cross–validation*

Akurasi 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 1 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*1:

Tabel 1. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 40% |
|  | 12 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |

Akurasi 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 2 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*2:

Tabel 2. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% |

Akurasi 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 3 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*4:

Tabel 3. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 20% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% |
| t | 3 | 20% | 0% | 0% | 60% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 20% | 0% |
|  | 11 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 40% |

Akurasi 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 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*5:

Tabel 4. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 20% | 0% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 40% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 20% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 40% | 20% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 11 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 20% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |

Akurasi 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 5 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*6:

Tabel 5. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 40% |
| t | 3 | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| u | 4 | 0% | 0% | 20% | 20% | 40% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 20% | 0% | 20% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
|  | 11 | 0% | 0% | 20% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 20% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% |

Akurasi 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 6 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*7:

Tabel 6. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 40% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 60% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 20% |

Akurasi 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 7 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*8:

Tabel 7. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
| t | 3 | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 20% |
|  | 11 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |

Akurasi 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 8 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*9:

Tabel 8. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 60% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 20% |
|  | 11 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% | 20% |

Akurasi 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 9 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*10:

Tabel 9. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 40% |
| t | 3 | 0% | 0% | 0% | 40% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 40% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 20% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |

**Lampiran D.5.** Disajikan Hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold cross–validation*

Akurasi 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). Pada Tabel 1 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*1:

Tabel 1. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 20% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 20% | 0% | 20% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 20% | 40% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |

Akurasi 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). Pada Tabel 2 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*2:

Tabel 2. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 20% | 0% | 0% |
| t | 3 | 0% | 0% | 20% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 60% |

Akurasi 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). Pada Tabel 3 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*4:

Tabel 3. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 20% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% |
| t | 3 | 20% | 0% | 0% | 40% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 40% | 20% | 20% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 20% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 40% |

Akurasi 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). Pada Tabel 4 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*5:

Tabel 4. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 40% | 0% | 0% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 40% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
|  | 7 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 60% | 0% | 20% | 0% | 0% | 0% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |

Akurasi 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). Pada Tabel 5 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*6:

Tabel 5. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| u | 4 | 0% | 0% | 0% | 20% | 40% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 20% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 20% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% |

Akurasi 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). Pada Tabel 6 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*7:

Tabel 6. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 60% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 20% |

Akurasi 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). Pada Tabel 7 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*8:

Tabel 7. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
| t | 3 | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 60% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |

Akurasi 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). Pada Tabel 8 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*9:

Tabel 8. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 20% |
|  | 11 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |

Akurasi 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). Pada Tabel 9 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*10:

Tabel 9. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 5 di *k–fold* iterasike*–*10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 20% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 40% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% |
|  | 11 | 0% | 0% | 20% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |

**Lampiran D.6.** Disajikan Hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold cross–validation*

Akurasi 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 1 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*1:

Tabel 1. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 20% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 40% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 40% |
|  | 12 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 20% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |

Akurasi 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 2 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*2:

Tabel 2. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% |
|  | 13 | 0% | 0% | 20% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |

Akurasi 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 3 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*3:

Tabel 3. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 40% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% | 0% |
|  | 11 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 20% |

Akurasi 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 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*4:

Tabel 4. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 20% | 0% | 20% | 0% |
| t | 3 | 20% | 0% | 0% | 40% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 20% | 0% | 40% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 60% |

Akurasi 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 5 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*5:

Tabel 5. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 40% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 20% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 40% | 0% | 0% |
|  | 7 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% | 20% |

Akurasi 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 6 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*6:

Tabel 6. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 20% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% |

Akurasi 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 7 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*7:

Tabel 7. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 60% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 40% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 20% |

Akurasi 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 8 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*9:

Tabel 8. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 20% | 0% | 0% |
| A | 1 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |
| t | 3 | 0% | 0% | 20% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 20% |
|  | 11 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 20% |

Akurasi 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 9 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*10:

Tabel 9. Akurasi Klasifikasi *Genre* Lagu menggunakan KNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 40% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 60% | 0% | 0% | 0% | 0% |
|  | 10 | 20% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |

**Lampiran D.7.** Disajikan Hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold cross–validation*

Akurasi 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). Pada Tabel 1 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*1:

Tabel 1. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 20% |
| t | 3 | 0% | 0% | 0% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 20% | 20% | 20% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 20% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |

Akurasi 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). Pada Tabel 2 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*2:

Tabel 2. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 40% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 20% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% |

Akurasi 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). Pada Tabel 3 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*3:

Tabel 3. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 40% |
| t | 3 | 0% | 0% | 0% | 80% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 20% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 60% | 0% | 0% |
|  | 7 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 80% | 0% | 0% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% | 20% |

Akurasi 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). Pada Tabel 4 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*4:

Tabel 4. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 20% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 20% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% |
| t | 3 | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 40% | 0% | 0% | 20% | 0% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 40% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 60% |

Akurasi 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). Pada Tabel 5 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*5:

Tabel 5. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 40% | 0% | 0% |
| A | 1 | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 40% | 0% | 20% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 40% | 20% | 0% |
|  | 7 | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 20% | 0% | 0% | 0% | 60% | 0% | 20% | 0% | 0% | 0% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 20% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% |

Akurasi 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). Pada Tabel 6 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*6:

Tabel 6. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 20% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| u | 4 | 0% | 0% | 0% | 20% | 60% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 20% |
|  | 11 | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 40% | 0% | 0% | 40% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 80% | 0% |
|  | 13 | 0% | 0% | 0% | 20% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% |

Akurasi 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). Pada Tabel 7 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*7:

Tabel 7. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% |
| A | 1 | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 20% | 20% | 0% | 0% |
| t | 3 | 0% | 0% | 0% | 40% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 20% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 0% | 0% |
|  | 11 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 60% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 20% | 20% |

Akurasi 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). Pada Tabel 8 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*9:

Tabel 8. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 40% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 20% | 0% | 20% | 0% | 0% |
| A | 1 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
| t | 3 | 0% | 0% | 20% | 60% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 20% | 0% | 20% | 0% |
|  | 7 | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% |
|  | 10 | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 40% | 20% | 20% |
|  | 11 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% |
|  | 13 | 0% | 0% | 20% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 20% | 0% | 20% |

Akurasi 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). Pada Tabel 9 disajikan hasil Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*10:

Tabel 9. Akurasi Klasifikasi *Genre* Lagu menggunakan MKNN pada *Confusion Matrix* saat nilai *k* = 7 di *k–fold* iterasike*–*10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prediksi | | | | | | | | | | | | | | | |
| \ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  | 0 | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% | 0% |
| A | 1 | 0% | 40% | 0% | 0% | 0% | 0% | 0% | 60% | 0% | 0% | 0% | 0% | 0% | 0% |
| k | 2 | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 20% | 0% | 20% |
| t | 3 | 0% | 0% | 0% | 40% | 40% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% |
| u | 4 | 0% | 0% | 0% | 40% | 60% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| a | 5 | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| l | 6 | 0% | 0% | 0% | 0% | 0% | 0% | 80% | 0% | 0% | 0% | 20% | 0% | 0% | 0% |
|  | 7 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% | 0% |
|  | 8 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 100% | 0% | 0% | 0% | 0% | 0% |
|  | 9 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 80% | 0% | 0% | 0% | 0% |
|  | 10 | 20% | 0% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 20% | 40% | 0% |
|  | 11 | 0% | 0% | 0% | 20% | 40% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |
|  | 12 | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 40% | 0% | 0% | 0% | 60% | 0% |
|  | 13 | 0% | 0% | 40% | 0% | 20% | 0% | 0% | 0% | 0% | 0% | 20% | 0% | 0% | 20% |

**Lampiran D.8.** Disajikan Hasil Perbandingan Akurasi pada perubahan nilai *k* PCA*–*MKNN dengan PCA*–*KNNdi Pengujian Kedua

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 1 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 21.10% | 22.90% | 22.60% | 23.90% | 23.30% | 23.60% | 23% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 23.40% | 23.70% | 24.90% | 25.70% | 25.60% | 25.10% | 25.60% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 25.90% | 25.90% | 25.70% | 25.30% | 24.60% | 25.10% | 24.90% |
| K=23 | K=24 | K=25 | K=26 | K=27 | K=28 | K=29 |
| 24.90% | 25.10% | 25.30% | 24.90% | 25% | 25.60% | 25.60% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 21.90% | 24.40% | 23.40% | 23.40% | 22.90% | 24.40% | 24.10% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 23.90% | 23.30% | 23.90% | 24.10% | 24.70% | 25.10% | 25.90% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 26% | 25.70% | 25.90% | 25.60% | 25% | 25.10% | 24.40% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 24.70% | 24.70% | 25% | 25.30% | 25.10% | 25.30% | 26.30% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 2 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 21.90% | 25.70% | 27.90% | 28.70% | 28.60% | 29.40% | 28.90% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 28.90% | 29.30% | 29.10% | 29% | 28.90% | 27.90% | 27.30% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 28.70% | 28.70% | 29.30% | 29% | 28.70% | 28.30% | 27% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 27.60% | 27.70% | 27.40% | 27.40% | 27.90% | 27.90% | 28.10% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 24.10% | 25.90% | 26.40% | 28.70% | 27.30% | 28% | 28.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 28.90% | 31.10% | 30.60% | 29.70% | 29% | 28.10% | 27.60% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 27.30% | 27.10% | 28.40% | 28.70% | 28.30% | 28.30% | 27.70% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 28.60% | 27.90% | 27% | 27.60% | 27.90% | 27.30% | 27.60% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 3 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 30.40% | 35.60% | 36.60% | 37.60% | 37.70% | 38.40% | 37.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 38% | 37% | 36.70% | 37.10% | 36.30% | 36.60% | 35.70% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 35.90% | 36% | 36.40% | 37% | 36.60% | 36.60% | 36.70% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 35.90% | 35.90% | 36% | 35.70% | 35.60% | 35.70% | 35.90% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 31.10% | 33% | 35.40% | 37.40% | 37.30% | 37% | 38.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 38.70% | 38% | 39.60% | 38.70% | 38.40% | 37.60% | 37.30% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 36.70% | 36.90% | 38% | 38.90% | 37.60% | 37.30% | 36.70% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 37.10% | 36.70% | 36.60% | 36.60% | 36.10% | 35.60% | 36% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 4 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 38.10% | 42% | 45.10% | 44.40% | 43.60% | 43.70% | 43.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 43.40% | 43.40% | 43.40% | 42.40% | 43.10% | 42.90% | 42.40% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 41.70% | 41.40% | 40.90% | 41.10% | 41.70% | 41.70% | 41.10% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 40.60% | 40.30% | 40.30% | 39.60% | 39.30% | 38.70% | 39.70% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 33.40% | 39.10% | 41.60% | 41.70% | 41.90% | 43.10% | 41.90% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 43% | 43.60% | 42.30% | 42.70% | 43.10% | 43% | 43.60% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 42.90% | 42.90% | 42.90% | 42.10% | 42.60% | 41.90% | 41.90% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 41% | 40.60% | 41% | 40.70% | 39.30% | 39.60% | 40.60% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 5 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 45.10% | 50.30% | 52.10% | 52% | 50.40% | 49.90% | 49.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 50.70% | 50.30% | 49.60% | 49% | 46.90% | 47.60% | 47% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 46.40% | 46.40% | 45.90% | 45.90% | 45% | 45% | 45.30% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 44.10% | 44.30% | 44.30% | 43.90% | 43.30% | 43% | 42.70% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 42.60% | 47.10% | 49.60% | 50.90% | 50.40% | 51.40% | 50.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 51.30% | 51.90% | 51% | 50% | 49.70% | 48.60% | 48% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 49% | 48.70% | 47.10% | 47.90% | 46.30% | 46.60% | 45.60% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 45.90% | 46.10% | 46% | 46.40% | 47.40% | 47.40% | 47% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 6 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 47.10% | 53% | 53% | 52.30% | 52.30% | 52.70% | 52.90% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 52% | 51.70% | 50.30% | 49.70% | 50% | 48.70% | 48.30% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 48% | 47.40% | 48.70% | 47.60% | 47.70% | 47.10% | 46.70% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 46.30% | 46% | 47% | 47% | 46.30% | 46.40% | 45.70% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 45.90% | 51.30% | 50.60% | 52% | 52.10% | 52.60% | 53.30% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 54.30% | 53.70% | 52.60% | 51.60% | 51.30% | 50.60% | 50.60% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 49.90% | 50.30% | 50.40% | 49.60% | 49.10% | 49% | 48.70% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 48.70% | 48.40% | 48.60% | 48.40% | 48.10% | 47.90% | 48.40% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 7 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 50% | 53% | 51.90% | 54.10% | 54.30% | 54.60% | 53.60% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 52.10% | 52.40% | 51.10% | 51.70% | 51.70% | 52% | 51.40% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 50.60% | 50.10% | 50% | 50% | 49.10% | 48.90% | 49.30% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 48.70% | 48% | 47.60% | 48.10% | 48.30% | 48.30% | 47.60% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 49.10% | 50.70% | 50.60% | 53.60% | 52.60% | 53% | 55.30% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 54.30% | 54.60% | 53.90% | 53.10% | 52.70% | 52.40% | 52.90% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 51.70% | 51.70% | 51% | 51.90% | 51.30% | 51.40% | 51.10% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 52% | 52% | 51.30% | 51.30% | 50.70% | 50.70% | 50.10% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 8 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 49% | 54.40% | 52.90% | 53.90% | 54.30% | 54.90% | 54.40% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 54.10% | 53% | 53.60% | 53.10% | 53.40% | 52.90% | 52.60% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 52.30% | 52.70% | 52.40% | 51.70% | 51.40% | 50.40% | 50% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 49.60% | 49.60% | 49.40% | 49% | 49.40% | 49.30% | 49.90% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 50.30% | 53.40% | 52.10% | 53.60% | 54.30% | 55.30% | 55.70% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 56% | 54.70% | 54.40% | 54.60% | 54.90% | 55.70% | 54.70% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 54.70% | 54.90% | 54.40% | 53.90% | 52.40% | 53.40% | 52.30% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 53% | 51.70% | 53.10% | 52.70% | 52.40% | 52.70% | 52% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 9 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 54.10% | 57% | 57.30% | 57.90% | 58.70% | 57.60% | 56.90% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 56.70% | 56% | 54.70% | 54.90% | 55.90% | 55.10% | 54.40% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 54.70% | 53.90% | 54% | 53.40% | 52.90% | 52.90% | 52.90% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 52% | 51.70% | 51.10% | 51.10% | 50.70% | 50.40% | 50.10% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 55% | 57.30% | 56.70% | 58.30% | 57.90% | 58.40% | 58.40% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 58% | 57.60% | 56.90% | 58% | 57.60% | 58.10% | 57.30% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 56.60% | 56.70% | 56.30% | 55.90% | 55.90% | 55.40% | 54.10% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 55.40% | 55.30% | 54.90% | 55% | 55.30% | 54.40% | 54.30% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 10 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 54% | 56.60% | 58.70% | 58% | 59.40% | 58.90% | 57.90% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 57.30% | 56.40% | 56.40% | 57% | 56.40% | 55.40% | 55.10% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 55.60% | 54.60% | 56% | 54% | 54.30% | 53.70% | 53.30% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 53% | 53% | 53% | 52.60% | 52% | 52.10% | 51% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 55.60% | 58.90% | 57.60% | 58.90% | 58.40% | 60% | 59% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 59.10% | 58.40% | 59.30% | 59.40% | 57.90% | 58.30% | 58.60% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 58.70% | 58% | 57.90% | 57.70% | 56.60% | 54.90% | 56.40% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 56.30% | 56.70% | 56.70% | 56.30% | 56.10% | 55.70% | 55.70% |

1. Metode PCA*–*MKNN dengan PCA*–*KNNpada 11 fitur baru.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Metode | Nilai k | | | | | | |
| PCA-MKNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 55.70% | 60% | 60.70% | 60% | 60.70% | 60.70% | 60.40% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 59.40% | 58.90% | 58.40% | 58.60% | 58.30% | 58.30% | 59% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 58.70% | 57.60% | 57.10% | 56.40% | 56.10% | 55.70% | 55% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 54.60% | 54% | 53.70% | 53.30% | 52.90% | 52.60% | 52.30% |
| PCA-KNN | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 |
| 58.10% | 60.70% | 61.70% | 63% | 61.70% | 63.10% | 61.90% |
| k=9 | k=10 | k=11 | K=12 | k=13 | k=14 | k=15 |
| 62% | 60.60% | 61.70% | 61.40% | 61% | 61% | 61.10% |
| k=16 | k=17 | k=18 | k=19 | k=20 | k=21 | k=22 |
| 61.30% | 60.60% | 59.70% | 59.90% | 59.40% | 60% | 58.90% |
| k=23 | k=24 | k=25 | k=26 | k=27 | k=28 | k=29 |
| 59.10% | 59.30% | 58.70% | 58.70% | 58.60% | 57.90% | 57.30% |