**TUGAS MANDIRI**
**(Fundamental Of Data Mining)**

# ANALISIS DAN PREDIKSI POPULARITAS LAGU SPOTIFY MENGGUNAKAN ALGORITMA RANDOM FOREST


**Nama : Indah Agusliani**
**NPM : 231510059**
**Dosen : Erlin Elisa, S.Kom., M.Kom.**



**PROGRAM STUDI SISTEM INFORMASI**
**FAKULTAS TEKNIK DAN KOMPUTER**
**UNIVERSITAS PUTERA BATAM**

**2026**

1. **Deskripsi Dataset**
    - Sumber dataset : Kaggle.com https://www.kaggle.com/datasets/wardabilal/spotify-global-music-dataset-20092025
    - Jumlah record : 8.582 record data lagu Spotify
    - Jumlah atribut : 16 atribut yang merepresentasikan informasi terkait lagu, artis, dan album
    - Tipe data :
        1) Numerik (integer dan float), seperti track_popularity, artist_popularity, artist_followers, dan track_duration_min
        2) Kategorial (object/boolean), seperti track_name, artist_name, album_type, dan explicit.
    - Target/label (jika supervised)
      popularitas lagu, yang dibagi menjadi dua kelas:

        1) 1 (Populer): jika track_popularity ≥ 50
        2) 0 (Tidak Populer): jika track_popularity < 50

    - Permasalahan yang ingin diselesaikan
      Dataset ini terdiri dari data lagu Spotify dengan berbagai atribut numerik dan kategorikal serta satu label klasifikasi untuk memprediksi popularitas lagu menggunakan algoritma Random Forest.

2. **Persiapan Data & Preprocessing**
    - Data Cleaning (Missing Value dan Outlier)

      Langkah pertama adalah melakukan pengecekan terhadap missing value pada dataset menggunakan fungsi isnull().sum(). Berdasarkan hasil pengecekan, terdapat beberapa data yang memiliki nilai kosong (NaN), khususnya pada atribut kategorikal seperti genre artis. Untuk mengatasi hal tersebut, dilakukan pembersihan data dengan menghapus baris data yang memiliki nilai kosong menggunakan metode dropna(). Pada penelitian ini tidak dilakukan penghapusan outlier secara khusus, karena algoritma Random Forest cukup robust terhadap keberadaan outlier.

    - Encoding Data Kategorikal

      Pada penelitian ini, tidak dilakukan encoding data kategorikal menggunakan LabelEncoder maupun OneHotEncoder. Hal ini dikarenakan model hanya menggunakan atribut numerik sebagai fitur input, sehingga atribut kategorikal tidak disertakan dalam proses pelatihan model.

- Scaling / Normalization

  Proses scaling atau normalisasi data tidak dilakukan pada penelitian ini. Hal ini disebabkan karena algoritma Random Forest tidak sensitif terhadap perbedaan skala data**,** sehingga performa model tetap stabil tanpa normalisasi.

- Feature Selection / Feature Engineering

  Atribut numerik yang digunakan:

  1) Popularitas artis
  2) Jumlah pengikut artis
  3) Durasi lagu
  4) Atribut numerik lainnya

  Selain itu, dilakukan feature engineering berupa pembuatan label klasifikasi popularity_label berdasarkan nilai track_popularity.

- Split Data Train dan Test

  1) 80% data latih (training data)
  2) 20% data uji (testing data)

  Pembagian data dilakukan menggunakan metode train_test_split dengan tujuan untuk menguji performa model terhadap data yang belum pernah dilihat sebelumnya.

| | Keterangan | Sebelum Preprocessing | Sesudah Preprocessing |
|---|---|---|---|
| 0 | Jumlah Record | 8582 | 5221 |
| 1 | Jumlah Atribut | 16 | 16 |
| 2 | Missing Value | Ada | Tidak ada |
| 3 | Target Label | Belum ada | popularity_label |

**Tabel 1. Ringkasan Dataset Sebelum dan Sesudah Preprocessing**

| | Jenis Data | Jumlah Data | Persentase |
|---|---|---|---|
| 0 | Data Latih (Train) | 6865 | 80% |
| 1 | Data Uji (Test) | 1717 | 20% |
| 2 | Total | 8582 | 100% |

**Tabel 2. Distribusi Data Train dan Test**

### 3. Analisis Statistik & Visualisasi

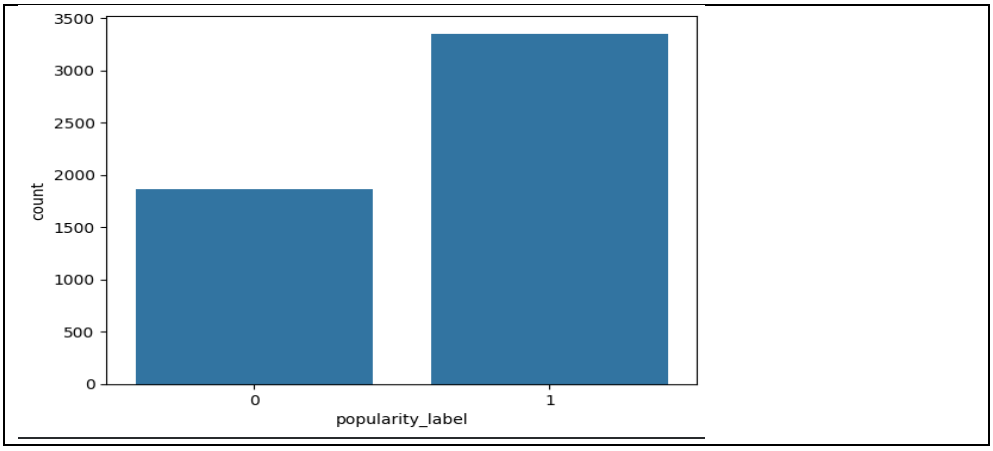
- Statistik deskriptif dataset

Statistik deskriptif diperoleh menggunakan fungsi describe() pada dataset. Nilai track_popularity memiliki rentang yang cukup lebar, menunjukkan variasi tingkat popularitas lagu. Atribut artist_followers memiliki nilai yang sangat besar dan tidak merata, menandakan adanya perbedaan signifikan antara artis populer dan kurang populer. Durasi lagu (track_duration_min) relatif stabil dan berada pada rentang waktu yang wajar untuk lagu pada umumnya.

| | track_number | track_popularity | artist_popularity | artist_followers | album_total_tracks | track_duration_min | popularity_label |
|---|---|---|---|---|---|---|---|
| count | 5221.000000 | 5221.000000 | 5221.000000 | 5.221000e+03 | 5221.000000 | 5221.000000 | 5221.000000 |
| mean | 6.465237 | 53.555258 | 74.588776 | 3.383704e+07 | 15.893124 | 3.564999 | 0.642406 |
| std | 6.324842 | 22.987208 | 17.178821 | 4.472907e+07 | 13.101610 | 1.133062 | 0.479338 |
| min | 1.000000 | 0.000000 | 2.000000 | 4.000000e+00 | 1.000000 | 0.140000 | 0.000000 |
| 25% | 1.000000 | 41.000000 | 63.000000 | 1.303279e+06 | 10.000000 | 2.900000 | 0.000000 |
| 50% | 5.000000 | 59.000000 | 79.000000 | 1.096042e+07 | 14.000000 | 3.510000 | 1.000000 |
| 75% | 10.000000 | 71.000000 | 88.000000 | 4.771699e+07 | 19.000000 | 4.080000 | 1.000000 |
| max | 102.000000 | 99.000000 | 100.000000 | 1.455421e+08 | 181.000000 | 13.510000 | 1.000000 |

- Distribusi Target / Label

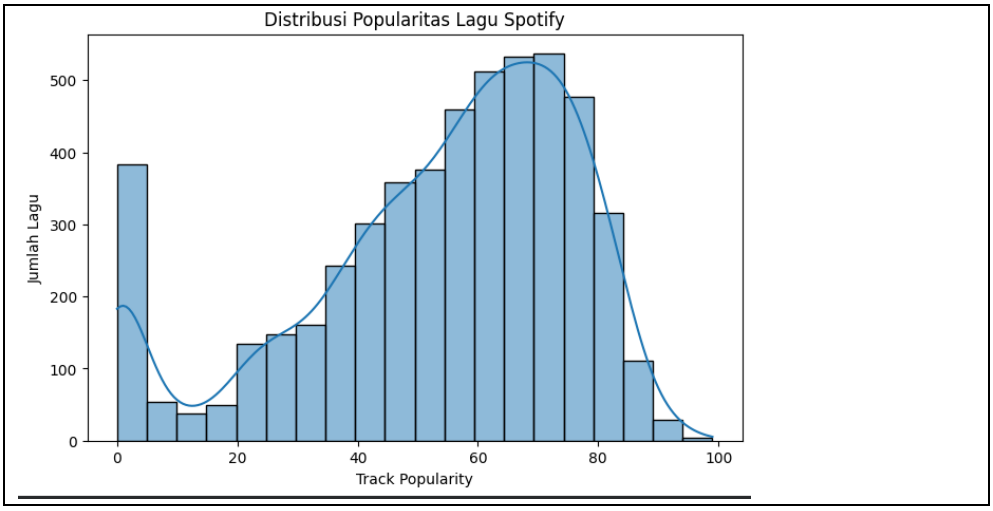
Distribusi target label divisualisasikan menggunakan grafik batang untuk melihat perbandingan jumlah lagu populer dan tidak populer**.** Distribusi data menunjukkan bahwa jumlah lagu tidak populer lebih banyak dibandingkan lagu populer**.** Kondisi ini mengindikasikan adanya ketidakseimbangan kelas (imbalanced dataset)**,** namun masih dapat ditangani dengan algoritma Random Forest.




- Korelasi Antar Fitur (Heatmap)

  Atribut artist_popularity dan artist_followers memiliki korelasi positif terhadap popularitas lagu. Tidak ditemukan korelasi yang sangat tinggi antar fitur input, sehingga risiko multikolinearitas relatif rendah. Fitur durasi lagu memiliki korelasi yang lemah terhadap popularitas lagu.

- Histogram




Berdasarkan grafik histogram distribusi popularitas lagu Spotify, dapat diamati bahwa sebagian besar lagu memiliki nilai popularitas yang relatif rendah**,** sedangkan hanya sebagian kecil lagu yang mencapai tingkat popularitas tinggi. Hal ini menunjukkan bahwa tidak semua lagu memiliki peluang yang sama untuk menjadi populer, dan popularitas cenderung terkonsentrasi pada lagu-lagu tertentu saja.

Distribusi data yang tidak merata (skewed) ini mengindikasikan adanya ketidakseimbangan kelas antara lagu populer dan tidak populer. Kondisi tersebut menjadi tantangan dalam proses klasifikasi, namun masih dapat ditangani dengan baik oleh algoritma Random Forest karena kemampuannya dalam menangani distribusi data yang tidak seimbang serta pola non-linear.

Selain itu, grafik histogram memperlihatkan bahwa rentang nilai popularitas cukup luas**,** sehingga atribut popularitas memiliki variasi yang memadai untuk dijadikan target dalam proses klasifikasi.

## 4. Pemilihan dan Penerapan Algoritma

Tuliskan:

- Nama algoritma : Random Forest
- Alasan pemilihan

  Algoritma Random Forest dipilih karena cocok digunakan untuk masalah klasifikasi dengan karakteristik data yang tidak linear dan memiliki kemungkinan outlier. Selain itu, Random Forest memiliki keunggulan dalam:

  1) Menangani dataset dengan distribusi kelas yang tidak seimbang
  2) Mengurangi risiko overfitting dibandingkan decision tree tunggal
  3) Mampu bekerja dengan baik tanpa memerlukan normalisasi data

  Pada penelitian ini, Random Forest digunakan untuk mengklasifikasikan popularitas lagu Spotify ke dalam dua kelas, yaitu populer dan tidak populer.

- Parameter utama yang digunakan

  1) n_estimators : jumlah pohon keputusan yang digunakan
  2) random_state : digunakan untuk menjaga konsistensi hasil eksperimen
  3) Parameter lain menggunakan nilai default dari library scikit-learn

📌 Algoritma yang diuji:

| Algoritma | Library Python | Tujuan |
|---|---|---|
| Random Forest | sklearn.ensemble | Klasifikasi & feature importance |

**5. Pengujian dan Evaluasi Model**

Metode Evaluasi:

| Jenis Tugas | Metrics Evaluasi |
|---|---|
| Klasifikasi | Accuracy, Precision, Recall, F1-Score, Confusion Matrix |

Tabel Hasil Klasifikasi:

| Algoritma | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Random Forest | 1.00 | 1.00 | 1.00 | 1.00 |

Berdasarkan hasil evaluasi, model Random Forest menghasilkan nilai accuracy, precision, recall, dan F1-score sebesar 1.00. Hasil ini menunjukkan bahwa model mampu mengklasifikasikan seluruh data uji dengan benar. Nilai evaluasi yang sangat tinggi ini kemungkinan dipengaruhi oleh pemilihan fitur yang sangat relevan terhadap target serta karakteristik dataset yang relatif mudah dipisahkan. Meskipun demikian, diperlukan evaluasi lanjutan menggunakan dataset yang lebih beragam untuk memastikan kemampuan generalisasi model.

**6. Analisis & Interpretasi Hasil**

Algoritma paling optimal ialah Random Forest, karena menghasilkan nilai accuracy, precision, recall, dan F1-score sebesar 1.00 serta mampu menangani data non-linear. Fitur paling berpengaruh Artist popularity**,** artist followers, dan track duration**.** Model sudah sangat baik, namun evaluasi terbatas pada satu dataset. Tidak terjadi underfitting, tetapi terdapat potensi overfitting karena hasil evaluasi mencapai 100%. Popularitas lagu Spotify lebih dipengaruhi oleh popularitas dan jumlah pengikut artis dibandingkan karakteristik lagu itu sendiri.

## 7. Kesimpulan & Rekomendasi

Kesimpulan:
Penelitian ini berhasil mencapai tujuan untuk memprediksi popularitas lagu Spotify. Algoritma Random Forest menjadi model terbaik karena menghasilkan performa klasifikasi yang sangat baik dan mampu menangani data non-linear.

Rekomendasi:
Untuk pengembangan selanjutnya, disarankan menambah jumlah data, melakukan hyperparameter tuning, menerapkan teknik balancing kelas jika data tidak seimbang, serta mencoba algoritma lain seperti KNN atau SVM untuk perbandingan performa.