

**MODEL PEMBELAJARAN DAN LAPORAN AKHIR
PROJECT-BASED LEARNING
MATA KULIAH BIG DATA
KELAS C**



**“Analisis Data Real Time Tren Populer Lagu Spotify Tahun 2023
Menggunakan Elasticsearch”**

DISUSUN OLEH KELOMPOK “III” :

- | | |
|-------------------------|---------------------------|
| 1. Rizki Amanda | (22083010045) - KETUA |
| 2. Adelia Yuandhika | (22083010066) - ANGGOTA |
| 3. Nezalfa Sabrina | (22083010067) - ANGGOTA |
| 4. Jasmine Aulia | (22083010074) - ANGGOTA |
| 5. Cahya Eka Melati | (22083010090) - ANGGOTA |
| 6. Zarqa Ananda Rakhman | (2399200064) - ANGGOTA |

DOSEN PENGAMPU:

TRESNA MAULANA FAHRUDIN, S.ST., MT (199305012022031007)

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAWA TIMUR
2024

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	3
1.3 Tujuan.....	3
1.4 Manfaat.....	3
BAB II.....	4
TINJAUAN PUSTAKA.....	4
2.1 Teori Penunjang.....	4
2.1.1 Spotify.....	4
2.1.2 Spotify Web API.....	4
2.1.3 Analisis Data Realtime.....	5
2.1.4 Python.....	6
2.1.5 Elasticsearch.....	6
2.2 Penelitian Terkait.....	7
BAB II.....	9
METODOLOGI PENELITIAN.....	9
3.1 Identifikasi masalah.....	9
3.2 Studi literatur.....	9
3.3 Scraping Data Spotify.....	9
3.4 Pra-Pemrosesan Data.....	10
3.5 Analisis dan Visualisasi Data.....	10
BAB IV.....	11
HASIL DAN PEMBAHASAN.....	11
4.1 Scraping Data Real Time from Spotify.....	11
4.1.1 Kode python.....	12
4.2 Inisialisasi ElasticSearch.....	16
4.2.1 Import Library.....	18
4.2.2 Import Modul.....	18
4.2.3 Data Collection.....	19
4.2.4 Ubah DataFrame menjadi format JSON.....	19
4.2.5 Mengindeks setiap data dari sebuah DataFrame ke dalam Elasticsearch.....	20
4.2.6 Potongan script untuk Pemeriksaan Data Quality.....	22
BAB V.....	23
KESIMPULAN.....	23
DAFTAR PUSTAKA.....	24
LAMPIRAN.....	25

DAFTAR GAMBAR

Gambar 1. <i>Logo Aplikasi Spotify</i>	4
Gambar 2. <i>Logo Spotify Web API</i>	4
Gambar 3. <i>Logo Bahasa Pemrograman Python</i>	6
Gambar 4. <i>Logo Aplikasi Elasticsearch</i>	6
Gambar 5. <i>Desain sistem Penelitian</i>	9
Gambar 6. <i>Web Spotify for Developers</i>	11
Gambar 7. <i>Dashboard Spotify for Developers</i>	11
Gambar 8. <i>Basic Information Otorisasi</i>	12
Gambar 9. <i>Data hasil Scrapping Spotify 2023</i>	15
Gambar 10. <i>ElastisSearch Create Deployment</i>	16
Gambar 11. <i>ElastisSearch Deployment</i>	17
Gambar 12. <i>Visualiasai Release Songs</i>	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Spotify merupakan platform *streaming* asal Swedia yang memberikan layanan hiburan berupa akses ke jutaan lagu dan podcast dari kreator yang ada di seluruh dunia. Melalui Spotify, pengguna dapat mendengarkan musik dan podcast secara online dengan gratis. Selain itu, apabila berlangganan premium, layanan Spotify dapat dinikmati secara offline dan bebas iklan. Spotify menawarkan berbagai fitur, antara lain menemukan musik atau podcast yang diinginkan, mengikuti artis favorit, dan membuat playlist (<https://support.spotify.com>). Dalam beberapa tahun terakhir, Spotify telah berkembang menjadi layanan streaming yang sangat populer di berbagai belahan dunia. Spotify mencatat bahwa selain memiliki 602 juta pengguna saat ini, layanan tersebut kini memiliki lebih dari 236 juta pengguna berbayar. Dengan bertambahnya jumlah pengguna, volume data yang dikelola oleh Spotify juga semakin berlimpah. Data yang dikumpulkan dan diproses dalam Spotify memiliki ukuran, distribusi, serta keragaman yang sangat besar, sehingga memenuhi kriteria untuk disebut sebagai big data (McKinsey, 2011). Dengan adanya data yang melimpah dari Spotify, penelitian ini bertujuan untuk mengidentifikasi lagu-lagu yang paling populer pada tahun 2023.

Popularitas lagu di Spotify dapat diukur melalui beberapa metrik kunci, antara lain jumlah streaming, yang menggambarkan seberapa sering sebuah lagu diputar; tingkat popularitas, yang dihitung oleh algoritma Spotify berdasarkan berbagai faktor termasuk interaksi pengguna seperti *likes*, *shares*, dan *comments*; serta frekuensi lagu masuk dalam playlist, yang menunjukkan seberapa sering lagu tersebut dimasukkan ke dalam daftar putar oleh pengguna. Jumlah streaming secara langsung mencerminkan tingkat keterlibatan pendengar dengan sebuah lagu, sedangkan tingkat popularitas memberikan gambaran yang lebih holistik dengan memperhitungkan berbagai bentuk interaksi pengguna yang lebih mendalam. Frekuensi masuk dalam playlist mengindikasikan daya tarik jangka panjang dan relevansi lagu dalam konteks mendengarkan musik sehari-hari. Menggunakan data ini, penelitian ini akan menganalisis tren dan pola yang muncul dalam preferensi musik pengguna Spotify, serta memahami faktor-faktor apa saja yang berkontribusi terhadap popularitas sebuah lagu.

Elasticsearch adalah alat yang banyak digunakan untuk pencarian dan analisis data secara real-time serta merupakan mesin pencarian dan analisis terdistribusi yang handal dan fleksibel. Sebagai komponen utama dari Elastic Stack, Elasticsearch menyimpan data secara terpusat, memungkinkan pencarian yang sangat cepat, relevansi yang dapat disesuaikan, dan analisis canggih yang dapat dengan mudah diskalakan (www.elastic.co). Dalam analisis data musik dari Spotify, Elasticsearch dapat memproses dan menganalisis data dalam jumlah besar secara efisien. Namun, penggunaan Elasticsearch juga menghadirkan berbagai tantangan, terutama dalam hal analisis real-time dan identifikasi faktor-faktor yang mempengaruhi popularitas lagu. Tantangan utama dalam menggunakan Elasticsearch untuk analisis data Spotify mencakup pengelolaan dan pemrosesan data skala besar, serta memastikan responsivitas dan keakuratan hasil analisis secara real-time. Selain itu, mengidentifikasi faktor-faktor yang mempengaruhi popularitas lagu membutuhkan data seperti karakteristik musik, genre, dan informasi artis. Proses ini memerlukan analisis yang komprehensif untuk mendapatkan wawasan yang bermakna. Dengan mengatasi tantangan ini, penelitian diharapkan dapat memberikan pemahaman yang signifikan tentang faktor-faktor yang mempengaruhi popularitas lagu di Spotify.

Berkaitan dengan hal-hal tersebut, dalam *Project Based Learning* (PJBL) mata kuliah Big Data, penulis mengusung proyek penelitian yang berjudul “Analisis Data Real Time Tren Populer Lagu Spotify Tahun 2023 Menggunakan Elasticsearch”. Proyek penelitian ini memanfaatkan Elasticsearch untuk menganalisis data lagu Spotify secara real time berdasarkan kepopuleran atau tren yang sedang berkembang. Elasticsearch akan menyimpan data dalam indeks yang dapat di-query secara real-time, memungkinkan pengguna untuk melakukan query terhadap data yang telah diindeks. Dengan Elasticsearch, data dapat disimpan, dicari, dan dianalisis secara efektif, mendukung berbagai kebutuhan analisis data real-time. Proyek penelitian ini memberikan pengetahuan mengenai implementasi big data dalam situasi nyata, khususnya industri musik digital. Selain itu, penelitian ini juga memberikan wawasan bagi para peneliti di bidang musikologi dan data science tentang dinamika popularitas musik di era digital. Penggunaan Elasticsearch pada proyek penelitian ini diharapkan dapat memberikan hasil yang relevan dan akurat. Diharapkan pula dapat ditemukan pola-pola tertentu dalam data yang akan bermanfaat bagi pengembang aplikasi musik, artis, dan industri musik secara umum dalam mengambil keputusan yang lebih baik.

1.2 Permasalahan

- a. Bagaimana menganalisis dan mengidentifikasi lagu-lagu yang paling populer di Spotify sepanjang tahun 2023 berdasarkan data seperti jumlah streaming, tingkat popularitas, dan frekuensi masuk dalam playlist pengguna?
- b. Bagaimana mengatasi tantangan penggunaan Elasticsearch untuk analisis real-time lagu dari Spotify dan mengidentifikasi faktor-faktor yang mempengaruhi popularitas lagu, termasuk karakteristik musik, genre, dan artis?

1.3 Tujuan

- Untuk mengetahui lagu yang paling populer pada tahun 2023 dan memberikan pemahaman mengenai lanskap musik digital di Spotify
- Untuk mengoptimalkan penggunaan Elasticsearch dalam analisis data real-time pada lagu Spotify dan mengidentifikasi faktor-faktor yang mempengaruhi popularitas lagu.

1.4 Manfaat

- Untuk Perkembangan Ilmu
Memberikan peluang bagi mahasiswa untuk mengembangkan berbagai keterampilan dan pengetahuan menggunakan Elasticsearch untuk analisis data real-time, serta memahami bagaimana pengguna berinteraksi dengan platform Spotify dan apa yang mendorong mereka untuk mendengarkan musik tertentu.
- Untuk Spotify
Memahami tren dan preferensi pengguna dapat membantu Spotify meningkatkan retensi pengguna dengan menawarkan konten yang lebih relevan dan menarik, menemukan peluang baru untuk monetisasi seperti penawaran iklan bertarget dan kemitraan dengan merek, serta meningkatkan pengalaman pengguna secara keseluruhan dengan menyempurnakan navigasi dan fitur platform.
- Untuk Industri Musik
Memahami tren pasar seperti, industri musik dapat menggunakan data real-time Spotify untuk memahami tren pasar, mengidentifikasi artis baru yang menjanjikan, dan membuat keputusan bisnis yang lebih tepat. Serta dapat mengembangkan strategi pemasaran yang lebih efektif.

BAB II

TINJAUAN PUSTAKA

2.1 Teori Penunjang

2.1.1 Spotify



Gambar 1. *Logo Aplikasi Spotify*

Spotify adalah layanan streaming musik digital yang sangat populer saat ini, menawarkan akses ke jutaan lagu, podcast, dan video dari berbagai artis di seluruh dunia(<https://support.spotify.com>). Ide tentang Spotify pertama kali dikemukakan oleh pendirinya, Daniel Ek dan Martin Lorentzon, pada tahun 2005 saat mereka berbagi musik di sebuah apartemen di Swedia. Versi beta Spotify diluncurkan pada tahun 2007, dan aplikasi resminya dirilis pada 7 Oktober 2008. Sejak itu, Spotify telah berkembang dan berekspansi ke lebih dari 56 negara, termasuk Amerika Serikat dan Filipina. Spotify telah menjalin kerjasama dengan Universal Music Group, Warner Music, EMI Group, Sony Music Entertainment, dan Merlin. Aplikasi Spotify dapat digunakan di berbagai perangkat digital seperti ponsel pintar, desktop, tablet, PlayStation, Xbox, smart TV, dan perangkat audio berbasis Bluetooth. Pengguna dapat mengunduh aplikasi Spotify dan memilih antara layanan berbayar (premium) atau gratis. Dengan membayar Rp. 49.900 per bulan, pengguna premium dapat mendengarkan musik secara offline, tanpa iklan, memilih lagu apa saja, dan mendapatkan kualitas audio yang lebih baik dibandingkan dengan layanan gratis (spotify.com, 2017).

2.12 Spotify Web API



Gambar 2. *Logo Spotify Web API*

Spotify Web API adalah pemrograman aplikasi yang disediakan oleh Spotify, yang memungkinkan pengembang untuk mengakses data dari katalog musik Spotify. Dengan API ini, pengembang dapat membangun aplikasi yang dapat memanfaatkan data seperti lagu, artis, album, dan playlist yang disimpan di Spotify. Data yang disediakan mencakup metadata tentang musik, termasuk informasi detail seperti judul lagu, artis, genre, dan banyak lagi. API ini bekerja berdasarkan prinsip REST (Representational State Transfer) yang sederhana, yang memungkinkan pengembang untuk mengirim permintaan dan menerima respon dalam format JSON. Melalui berbagai endpoint yang disediakan, seperti `user profile` dan `browse`, pengembang dapat mengambil informasi pengguna, rekomendasi playlist berdasarkan suasana hati atau aktivitas, dan informasi lainnya dari katalog data Spotify (Dylan Crawshaw, 2020). Dokumentasi lengkap dan petunjuk penggunaan API Spotify dapat diakses di [Spotify Developer](<https://developer.spotify.com>), yang menyediakan panduan mendetail tentang cara mengintegrasikan dan memanfaatkan API ini dalam aplikasi.

2.1.3 Analisis Data Realtime

Analisis data real-time adalah proses cepat dalam menganalisis data untuk memberikan wawasan yang berguna bagi perusahaan. Dengan menggunakan algoritma machine learning dan teknologi otomatis lainnya, data masukan diubah menjadi informasi yang dapat digunakan. Hal ini membantu perusahaan mengurangi risiko, menghemat biaya, dan memahami lebih dalam tentang karyawan, pelanggan, serta kondisi keuangan mereka. Melalui analisis ini, perusahaan dapat memprediksi kapan perangkat akan rusak, memberi peringatan sebelumnya, dan mengambil tindakan cepat berdasarkan data real-time yang terus diperbarui. Data untuk analisis real-time dikumpulkan dari berbagai sumber seperti sensor, aplikasi, basis data, dan media sosial, kemudian disimpan di platform terpusat untuk dianalisis. Hasilnya ditampilkan dalam bentuk visualisasi interaktif, dashboard, atau laporan yang mendukung pengambilan keputusan cepat (Bunga Dea Laraswati, 2023).

Meskipun analisis data real-time menawarkan banyak manfaat seperti visualisasi data langsung, pengambilan keputusan lebih cepat, peningkatan hubungan pelanggan, peningkatan logistik, adaptasi lebih baik terhadap fluktuasi pasar, dan operasional yang lebih lancar, terdapat juga tantangan yang harus diatasi. Tantangan tersebut termasuk kebutuhan akan pemrosesan data yang cepat, biaya yang signifikan, integrasi data yang rumit, kebutuhan akan tim profesional, dan keamanan data. Di

masa depan, analisis data real-time diprediksi akan semakin penting dalam berbagai industri, membantu perusahaan merespons perubahan pasar dengan cepat, meningkatkan efisiensi operasi, dan tetap kompetitif dalam lingkungan yang semakin didorong oleh data(blog.algorit.ma, 2023).

2.1.4 Python



Gambar 3. *Logo Bahasa Pemrograman Python*

Python adalah bahasa pemrograman yang sering digunakan untuk pengembangan situs web, perangkat lunak/aplikasi, otomatisasi tugas, dan analisis data. Python bersifat serba guna, yang berarti bisa digunakan untuk membuat berbagai macam program, bukan hanya untuk tujuan tertentu saja. Berkat fleksibilitas dan kemudahan penggunaannya, Python menjadi salah satu bahasa pemrograman paling populer, terutama di kalangan pemula. Menurut survei pengembang Stack Overflow tahun 2022, Python adalah bahasa pemrograman terpopuler keempat. Sekitar 50% responden menyatakan bahwa mereka menggunakan Python selama hampir setengah dari waktu kerja mereka. Nama Python terinspirasi dari Monty Python. Saat Guido van Rossum menciptakan bahasa ini, ia sedang membaca skrip Monty Python's Flying Circus dari BBC. Dia merasa nama tersebut singkat dan sedikit misterius, sehingga memilihnya sebagai nama untuk bahasa pemrograman yang ia buat (dicoding, 2023).

2.1.5 Elasticsearch



Gambar 4. *Logo Aplikasi Elasticsearch*

Elasticsearch adalah mesin pencari dan analisis data yang didistribusikan dan open-source untuk semua jenis data, termasuk tekstual, numerik, geospasial, terstruktur, dan tidak terstruktur. Dibangun di atas Apache Lucene dan pertama kali dirilis pada tahun 2010 oleh Elastic, Elasticsearch dikenal dengan REST API yang sederhana, sifat terdistribusi, kecepatan, dan skalabilitasnya (ICHI.PRO, 2020-2024). Elasticsearch mencari dan menganalisis data melalui proses Data Ingestion, di mana data mentah diproses melalui tahap penguraian, normalisasi serta diperkaya dengan data tambahan sebelum dilakukan proses pengindeksan. Selain itu Elasticsearch adalah alat yang bersifat scalable yaitu dapat disesuaikan tergantung pada kebutuhan sistem. Penggunaan Elasticsearch umumnya termasuk pencarian aplikasi, pencarian perusahaan, pencarian situs web, penyimpanan log, dan pengindeksan (Elang Putra Sartika & Andhik Budi Cahyono, 2020).

2.2 Penelitian Terkait

2.2.1 Khairunnisa Junaidi dan rekan-rekannya (2023) dalam penelitiannya yang berjudul “Implementasi MapReduce pada Dataset Spotify Top Music Untuk Mengetahui Artis yang Paling Banyak Didengar Dalam Kurun Waktu 10 Tahun”. Menganalisis dataset Spotify Top Music 2010-2022 menggunakan Hadoop, sebuah framework open source yang mendukung MapReduce serta Python, bahasa pemrograman yang populer dan mudah digunakan untuk analisis dan visualisasi data. Penelitian ini bertujuan untuk mengidentifikasi artis yang paling populer selama 10 tahun terakhir, serta memberikan informasi tentang tren musik dalam periode tersebut. Hasil penelitian ini menunjukkan Taylor Swift, Drake, Justin Bieber, Ariana Grande, dan Ed Sheeran, merupakan artis dengan jumlah unduhan terbanyak selama sepuluh tahun terakhir. Hasil tersebut dapat digunakan untuk berbagai tujuan seperti strategi pemasaran, penelitian industri musik, dan analisis tren. Namun dalam penelitian ini tidak mempertimbangkan demografi pendengar seperti usia, lokasi geografis, dan preferensi musik yang bisa memberikan wawasan lebih tentang popularitas artis.

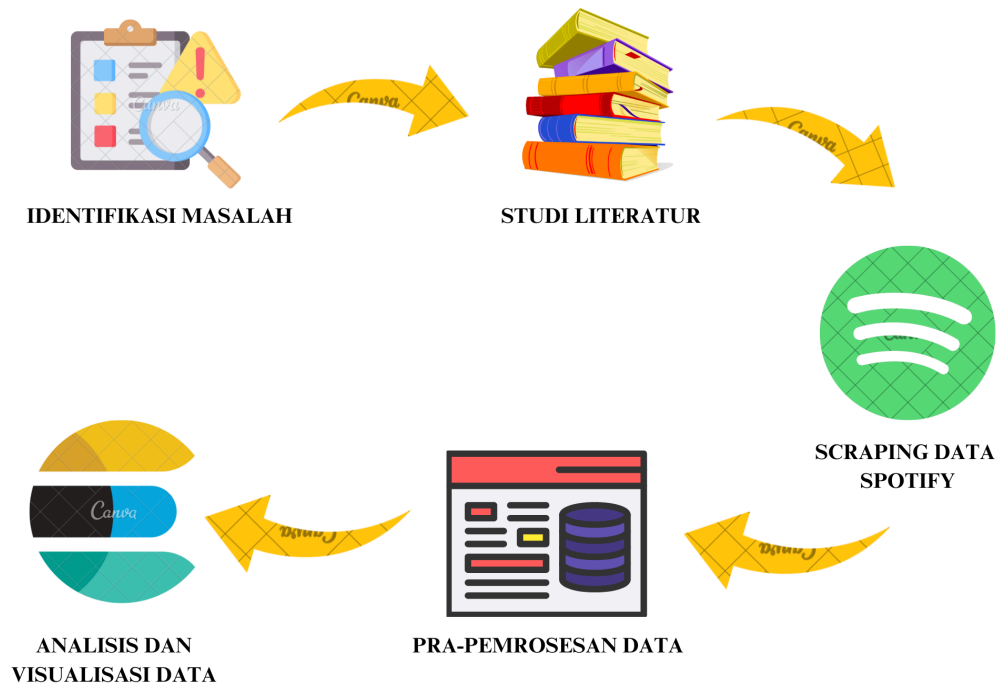
2.2.2 Penelitian "Prediksi Tren Musik Tahun 2024 pada Platform Spotify Menggunakan Metode Random Forest Regressor" oleh Hikmatul Kamilah dan rekan-rekannya (2023). Bertujuan untuk memprediksi tren musik tahun 2024 dengan menggunakan data lagu yang paling banyak didengarkan pada tahun 2023. Analisis dilakukan menggunakan Metode Random Forest Regressor untuk memprediksi

karakteristik musik yang kemungkinan akan menjadi tren di tahun 2024. Hasil penelitian menunjukkan bahwa algoritma Random Forest Regressor dapat digunakan untuk mengidentifikasi dan memprediksi musik yang kemungkinan akan menjadi tren di tahun 2024 berdasarkan data tren musik tahun 2023 di platform Spotify. Musik yang kemungkinan akan menjadi tren di tahun 2024 memiliki ciri-ciri seperti BPM tinggi, danceability tinggi, valence tinggi, energy tinggi, acousticness rendah, liveness rendah, dan speechiness sangat rendah.

2.2.3 Dwi Nuriska dan rekan-rekannya (2023) dalam penelitiannya yang berjudul “Klasterisasi Data Lagu Terpopuler Spotify 2023 Berdasarkan Suasana Hati Menggunakan Algoritma K-Means” menerapkan algoritma K-Means dalam analisis lagu Spotify 2023. Tujuan dari penelitian ini yaitu untuk mengelompokkan lagu berdasarkan suasana hati, sehingga pengguna dapat lebih mudah menemukan lagu yang sesuai dengan suasana hati mereka. Dalam penelitian ini, data lagu-lagu terpopuler Spotify 2023 akan dikategorikan berdasarkan empat kategori suasana hati, yaitu marah, sedih, rileks, dan bahagia, menggunakan algoritma K-Means Clustering dan pendekatan Knowledge Discovery in Database (KDD). Atribut-atribut audio seperti tempo, danceability, valence, dan energy digunakan dalam analisis ini. Hasil penelitian menunjukkan nilai Davies-Bouldin Index (DBI) terkecil sebesar 0,299 dengan jumlah $K = 3$, di mana cluster 0 terdiri dari lagu dengan suasana hati bahagia sebanyak 361 lagu, cluster 1 terdiri dari lagu dengan suasana hati rileks sebanyak 329 lagu, dan cluster 2 terdiri dari lagu dengan suasana hati sedih sebanyak 252 lagu. Tidak ditemukan cluster yang jelas menggambarkan suasana hati marah. Dengan distribusi anggota yang merata di setiap cluster, tidak ada suasana hati yang mendominasi dari hasil pengelompokan lagu.

BAB II

METODOLOGI PENELITIAN



Gambar 5. Desain sistem Penelitian

3.1 Identifikasi masalah

Identifikasi masalah menjadi tahap awal dari penelitian ini, yang mana berfokus pada proses penentuan tujuan dari penelitian ini yakni analisis tren populer lagu di Spotify tahun 2023 menggunakan Elasticsearch. Dengan melakukan identifikasi masalah ini secara sistematis, diharapkan penelitian ini dapat menghasilkan temuan yang relevan dan signifikan terkait tren musik di Spotify pada tahun 2023.

3.2 Studi literatur

Studi literatur merupakan tahap penting dalam penelitian, yang mana melibatkan review literatur yang relevan topik penelitian, seperti tren musik dan analisis data pada Spotify. Dalam proses ini, beberapa sumber literatur akan dikaji untuk membantu menentukan dan menemukan data yang tepat serta mengetahui proses analisis yang sesuai dengan permasalahan yang didapatkan.

3.3 Scraping Data Spotify

Tahap scraping data Spotify bertujuan untuk mengumpulkan data lagu dari platform Spotify menggunakan teknik scraping. Proses ini dilakukan dengan beberapa *package* seperti Selenium dan BeautifulSoup, untuk mengakses situs web Spotify dan mengambil

informasi yang diperlukan. Selain itu, dilakukan juga akses ke API Spotify untuk mendapatkan data yang lebih terstruktur, termasuk informasi tentang nama lagu, album, artis, dan karakteristik musik seperti danceability, acousticness, energi, dan lainnya.

3.4 Pra-Pemrosesan Data

Pra-Pemrosesan Data bertujuan untuk mempersiapkan data sebelum dilakukan analisis lebih lanjut. Proses ini melibatkan penggunaan bahasa pemrograman Python seperti menyiapkan format data yang sesuai untuk dimasukkan ke dalam Elasticsearch. Hal ini melibatkan pemilihan atribut yang relevan dari dataset yang telah diproses dan menyesuaikan skema data agar sesuai dengan kebutuhan analisis di Elasticsearch.

3.5 Analisis dan Visualisasi Data

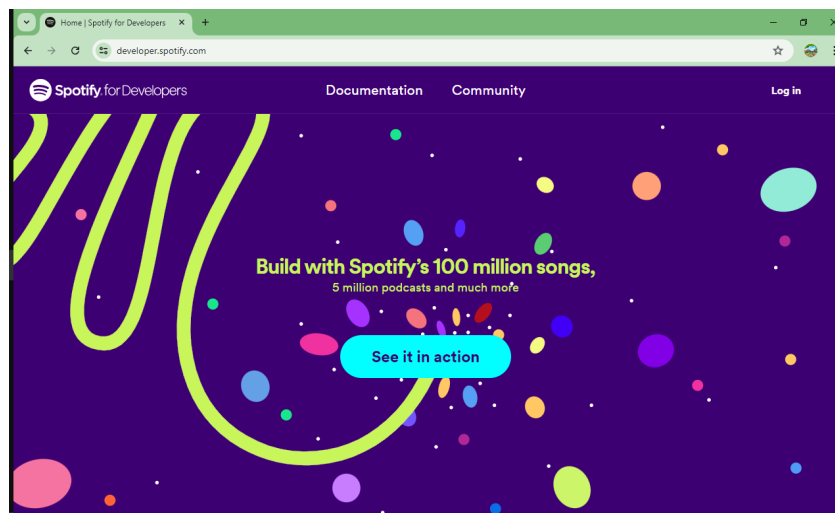
Pada tahap ini, melibatkan penggunaan Elasticsearch untuk analisis data dan pembuatan visualisasi. Dalam analisis ini mencakup identifikasi lagu-lagu populer berdasarkan metrik seperti jumlah streaming dan tingkat popularitas. Dengan melakukan analisis dan visualisasi data secara menyeluruh, penelitian ini diharapkan dapat memberikan wawasan yang mendalam tentang tren musik di Spotify pada tahun 2023 dan faktor-faktor yang mempengaruhi popularitas lagu.

BAB IV

HASIL DAN PEMBAHASAN

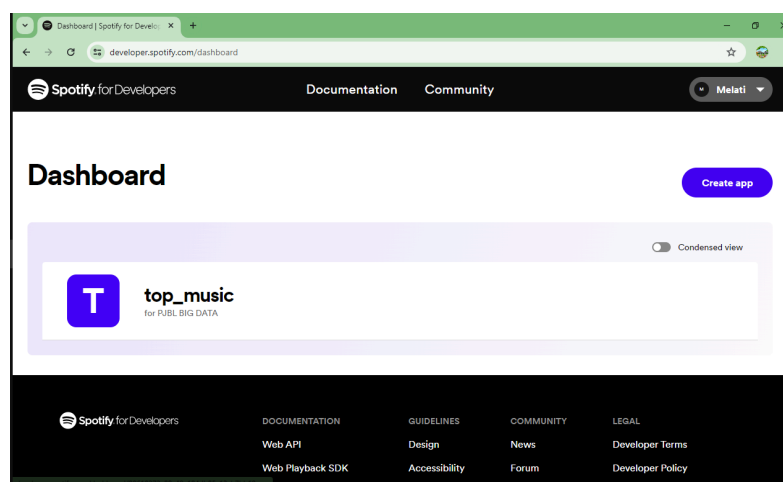
4.1 Scraping Data Real Time from Spotify

Data dari Spotify diakses melalui Spotify for Developers API. API ini menyediakan akses ke berbagai informasi yang dibutuhkan untuk menganalisis data real-time seperti informasi tentang lagu dan fitur-fitur dalam lagu.



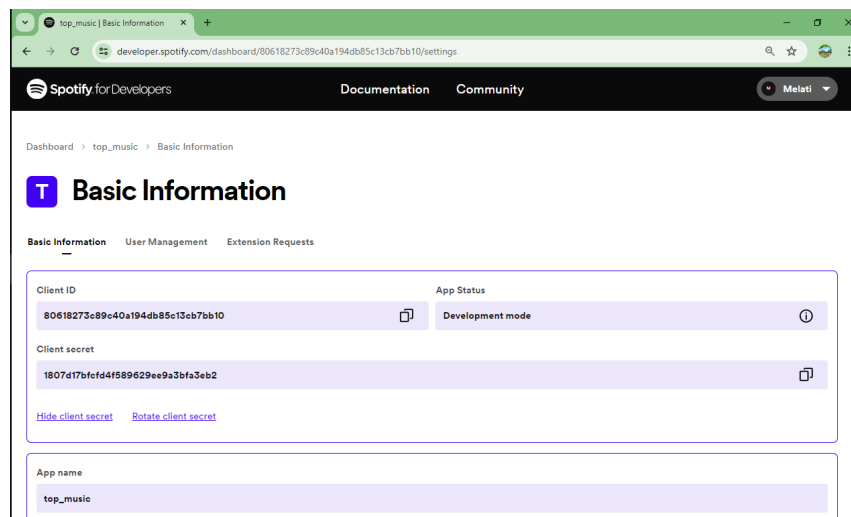
Gambar 6. Web Spotify for Developers

Untuk menggunakan Spotify Web API, kita perlu mendapatkan token akses yang akan digunakan untuk mengotentikasi permintaan data. Untuk mendapatkan token akses tersebut langkah pertama adalah membuat aplikasi di Spotify for Developers yang kami beri nama "top_music" dan mendapatkan Client ID dan Client Secret.



Gambar 7. Dashboard Spotify for Developers

Setelah itu, kita mendapatkan kode otorisasi dengan mengirimkan pengguna ke halaman otorisasi Spotify. Setelah pengguna memberikan izin, mereka akan dialihkan kembali ke aplikasi kita dengan kode otorisasi. Selanjutnya, kita dapat menukarkan kode otorisasi dengan token akses dengan mengirimkan permintaan POST ke endpoint token Spotify. Token akses ini akan digunakan dalam setiap permintaan API untuk mengotentikasi permintaan tersebut.



Gambar 8. Basic Information Otorisasi

Setelah itu, kita mendapatkan Client ID dan Client secret yang dimana ID ini digunakan untuk melakukan scraping data pada python. Client ID sendiri digunakan untuk mengidentifikasi aplikasi saat melakukan panggilan API ke Spotify sedangkan Client secret digunakan bersama Client ID untuk mengotentikasi aplikasi ke server Spotify.

4.1.1 Kode python

```
pip install spotipy
```

Kode skrip python di atas berfungsi untuk menginstal pustaka Python bernama “Spotipy” ke dalam sistem. Spotipy sendiri merupakan pustaka ringan yang memungkinkan untuk berinteraksi dengan API Web Spotify. Melalui “Spotipy” penulis dapat mengakses informasi tentang lagu, album, dan artis, seperti nama, genre, dan tanggal rilis.

```
import spotipy
from spotipy.oauth2 import SpotifyClientCredentials

client_id = '80618273c89c40a194db85c13cb7bb10'
client_secret = '1807d17bfcfd4f589629ee9a3bfa3eb2'

auth_manager = SpotifyClientCredentials(client_id=client_id,
client_secret=client_secret)
sp = spotipy.Spotify(auth_manager=auth_manager)
```

Kode tersebut digunakan untuk mengakses Spotify Web API menggunakan library Spotipy. Ini mencakup autentikasi dengan menggunakan `client_id` dan `client_secret` yang diperoleh dari Spotify for Developers, serta pembuatan objek Spotify untuk membuat permintaan API.

```
# menentukan atribut yg akan dipanggil

def get_features(tr_id):
    meta = sp.track(tr_id)
    features = sp.audio_features(tr_id)

    name = meta['name']
    album = meta['album']['name']
    artist = meta['album']['artists'][0]['name']
    release_date = meta['album']['release_date']
    length = meta['duration_ms']
    popularity = meta['popularity']

    acousticness = features[0]['acousticness']
    danceability = features[0]['danceability']
    energy = features[0]['energy']
    instrumentalness = features[0]['instrumentalness']
    liveness = features[0]['liveness']
    loudness = features[0]['loudness']
    speechiness = features[0]['speechiness']
    tempo = features[0]['tempo']
    time_signature = features[0]['time_signature']

    track = [name, album, artist, release_date, length, popularity,
             danceability, acousticness,
             energy, instrumentalness, liveness, loudness, speechiness,
             tempo, time_signature]

    return track
```

Selanjutnya adalah menentukan atribut-atribut yang ingin dipanggil dari katalog Spotify. Atribut-atribut ini dipilih berdasarkan kebutuhan untuk melakukan analisis tren populer lagu Spotify pada tahun 2023. Atribut-atribut yang dipilih antara lain, nama lagu, album, artis, tanggal rilis, durasi lagu, kepopuleran, *acousticness*, *danceability*, *energy*, *instrumentalness*, *liveness*, *loudness*, *speechiness*, *tempo*, dan *time_signature*. Dalam atribut-atribut tersebut termasuk didalamnya beberapa cara untuk mengukur aspek-aspek teknis dan musikal dari sebuah lagu menggunakan analisa audio, berikut penjelasannya.

a. Acousticness (Keakuistikan)

Nilai antara 0.0 dan 1.0 yang menunjukkan kemungkinan sebuah lagu didominasi oleh instrumen akustik. Nilai 0.0 menandakan lagu elektronik, sedangkan 1.0 menandakan lagu akustik, seperti lagu yang kebanyakan menggunakan gitar, piano, atau instrumen akustik lainnya.

b. Danceability (Dansa)

Nilai antara 0.0 dan 1.0 yang menunjukkan seberapa cocok lagu tersebut untuk berdansa. Nilai 0.0 berarti lagu tersebut kurang cocok untuk berdansa, sementara 1.0 berarti lagu tersebut sangat cocok untuk berdansa. Faktor yang mempengaruhi danceability termasuk tempo, ritme, dan kekuatan beat.

c. Energy (Energi)

Nilai antara 0.0 dan 1.0 yang menggambarkan intensitas dan aktivitas sebuah lagu. Nilai 0.0 berarti lagu tersebut tenang dan lembut, sementara 1.0 berarti lagu tersebut bersemangat dan penuh energi. Faktor yang mempengaruhi energi termasuk range dinamis, kekerasan suara, dan kehadiran instrumen perkusi.

d. Instrumentalness (Instrumental)

Nilai antara 0.0 dan 1.0 yang menunjukkan kemungkinan sebuah lagu hanya berisi instrumen, tanpa vokal. Nilai 0.0 berarti lagu tersebut pasti memiliki vokal, sedangkan nilai mendekati 1.0 berarti kemungkinan besar lagu tersebut instrumental.

e. Liveness (Keadaan Live)

Nilai antara 0.0 dan 1.0 yang menunjukkan kemungkinan sebuah lagu direkam secara live (langsung). Nilai 0.0 berarti kecil kemungkinan lagu tersebut direkam secara live, sedangkan nilai diatas 0.8 menandakan kemungkinan besar lagu tersebut direkam secara live.

f. Loudness (Kenyaringan)

Nilai dalam desibel (dB) yang menunjukkan keseluruhan kenyaringan sebuah lagu. Nilai ini berguna untuk membandingkan kenyaringan relatif antar lagu.

g. Speechiness (Bicara)

Nilai antara 0.0 dan 1.0 yang menunjukkan kemungkinan sebuah lagu mengandung suara bicara. Nilai 0.0 berarti lagu tersebut hanya berisi musik, sedangkan nilai diatas 0.66 menunjukkan kemungkinan besar lagu tersebut mengandung percakapan atau kata-kata yang diucapkan.

h. Tempo (Kecepatan)

Nilai dalam ketukan per menit (BPM) yang menunjukkan kecepatan lagu.

i. Time Signature (Tanda Waktu)

Estimasi pola ketukan secara keseluruhan dalam sebuah lagu. Tanda waktu ini ditulis dengan angka pecahan, misalnya 4/4 (waktu biasa) atau 3/4 (waltz).

```
def get_songs():
    tracks = []
    for i in range(0, 1000, 50):
        song = sp.search(q='year:2023', type='track', limit=50)
        for j in range(50):
            tr_id = song['tracks']['items'][j]['id']
            track = get_features(tr_id)
            tracks.append(track)

    df = pd.DataFrame(tracks, columns = ['name', 'album', 'artist',
    'release_date', 'length', 'popularity',
    'danceability', 'acousticness',
    'energy',
    'instrumentalness', 'liveness',
    'loudness', 'speechiness', 'tempo', 'time_signature'])
    return df
```

Selanjutnya adalah melakukan pengambilan informasi dan fitur lagu dari Spotify untuk tahun 2023. Kode skrip ini memanfaatkan *looping* atau perulangan untuk memperoleh data yang diinginkan. Proses *looping* yang dilakukan meliputi *looping* untuk mencari lagu dan *looping* untuk mengekstrak fitur dari lagu. Data yang telah berhasil diperoleh kemudian disimpan kedalam bentuk data frame. Baris kode terakhir yaitu `return df` berfungsi untuk mengembalikan data frame yang berisi informasi dan fitur lagu yang telah diekstrak dari Spotify.

```
df = get_songs()
df
```

Selanjutnya memanggil fungsi `'get_songs()'` untuk mendapatkan data lagu dan menyimpannya dalam DataFrame `'df'`. Kemudian, `'df'` digunakan untuk menampilkan data lagu tersebut dengan hasil seperti berikut:



	name	album	artist	release_date	length	popularity	danceability	acousticness	energy	instrumentalness	liveness
0	En Las Calles De Madrid	En Las Calles De Madrid	EDM NATIVES	2023-10-25	209400	0	0.4700	0.017800	0.970	0.899000	0.1210
1	BLACK JAM	BLACK JAM	NEO JAPONISM	2023-02-15	184153	6	0.3840	0.008230	0.954	0.000000	0.0572
2	Violin Concerto in D Major, RV 229: III. Allegro	Vivaldi: Concerti per violino XI 'Per Anna Maria'	Antonio Vivaldi	2023-09-22	198802	5	0.4020	0.916000	0.171	0.259000	0.1180
3	Bludsteroo	Sugma, Vol. 4	Friox	2023-11-19	161486	0	0.7180	0.116000	0.779	0.643000	0.1210
4	Durga Sapta Shloki	Sarveswari Durga Maa	Various Artists	2023-11-17	172997	0	0.4950	0.785000	0.145	0.000000	0.1110
5	My Pronouns	My Pronouns Are ..	Dubskie	2023-08-18	195813	7	0.7880	0.023900	0.829	0.000000	0.2080

Gambar 9. Data hasil Scrapping Spotify 2023

Pada Gambar 9. dapat dilihat data hasil scrapping Spotify tahun 2023 yang memiliki 15 kolom yang terdiri dari nama lagu ("name"), album ("album"), artis ("artist"), tanggal rilis ("release_date"), panjang lagu dalam detik ("length"), dan beberapa metrik lainnya seperti danceability, popularity, acousticness, energy, instrumentalness, dan lain-lainnya.

```
df.drop_duplicates()
```

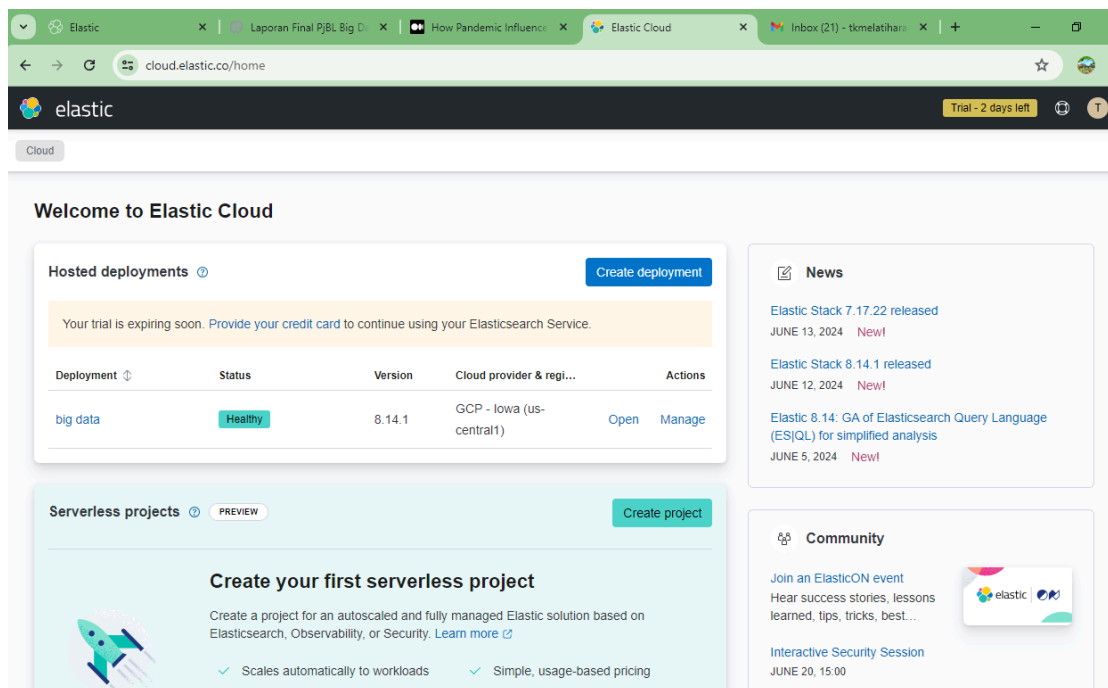
Selanjutnya menghapus baris yang duplikat, yaitu baris-baris yang memiliki nilai yang sama di semua kolom. Setelah baris duplikat dihapus, DataFrame df akan berisi data yang unik.

```
df.to_csv('final_topmusic_2023.csv', index=False)
```

Selanjutnya simpan DataFrame df ke dalam file CSV dengan nama 'final_topmusic_2023.csv'. Argumen index=False digunakan untuk menghilangkan penambahan indeks baris dalam file CSV yang dihasilkan.

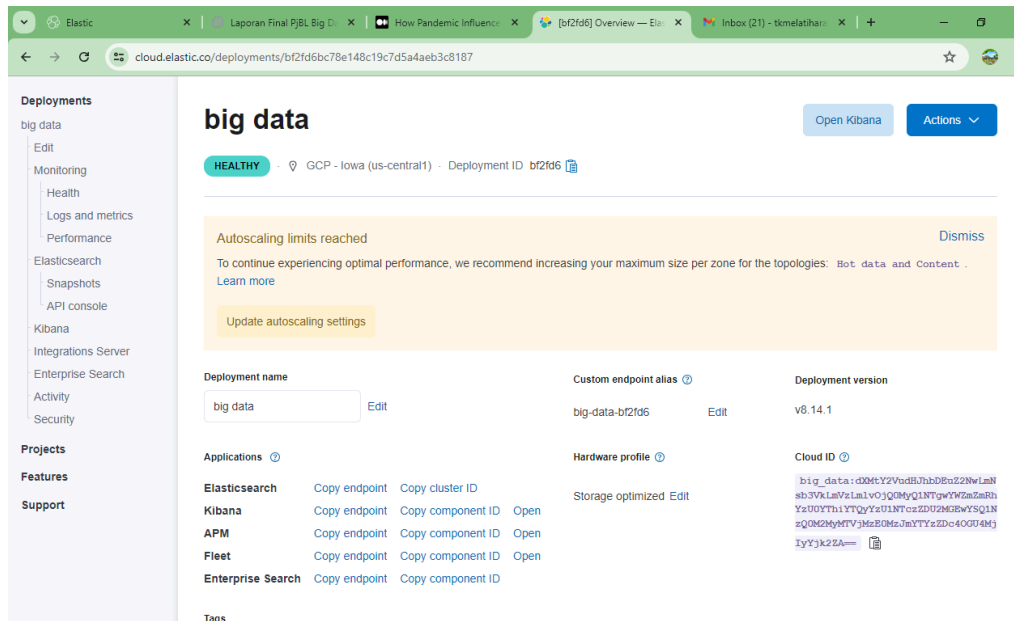
4.2 Inisialisasi ElasticSearch

Untuk melakukan analisis data menggunakan Elasticsearch di web cloud.elastic.co dengan menggunakan data Spotify 2023, kita perlu membuat deployment baru dengan nama "Big Data" di cloud.elastic.co. Setelah masuk, klik "Create deployment" dan pilih template "Elastic Stack (Elasticsearch, Kibana, and more)", lalu sesuaikan konfigurasi sumber daya dan buat deployment tersebut. Setelah deployment siap, catat cloud_id dan elastic_password yang diberikan.



Gambar 10. *ElasticSearch Create Deployment*

Berikut merupakan tampilan dari deployment “big data” pada cloud.elastic.co yang berisikan cloud_id dan elastic_password yang diberikan.



Gambar 11. ElasticSearch Deployment

Selanjutnya, masuk ke Python lalu instal library elasticsearch dengan menjalankan `!pip install elasticsearch`.

```
!pip install elasticsearch

from elasticsearch import Elasticsearch

# password for the 'elastic' user generated by Elasticsearch
elastic_pass = 'Z4DUHFQsx7z4I3ixxmIDMeH2'

# found in the 'manage deployment' page
cloud_id
='big_data:dXmTY2VudHJhbDEuZ2NwLmNsb3VkLmVzLmlvOjQ0MyQ1NTgwYWZmZmRhYzU0YThi
YTQyYzU1NTczZDU2MGEwYSQ1NzQ0M2MyMTVjMzE0MzJmYTZzZDc4OGU4MjIyYjk2ZA=='

# create the client instance
elastic = Elasticsearch(
    cloud_id=cloud_id,
    basic_auth=('elastic', elastic_pass)
)

# test the connection
if elastic.ping():
    print('connected to Elasticsearch')
else:
    print('connection failed')
```

Kode skrip tersebut digunakan untuk menghubungkan aplikasi Python ke instans Elasticsearch yang di-hosting di cloud. Pertama, pustaka elasticsearch diinstal menggunakan pip. Kemudian, variabel `elastic_pass` dan `cloud_id` didefinisikan untuk menyimpan kata sandi pengguna 'elastic' dan informasi kluster cloud. Selanjutnya, instans klien Elasticsearch dibuat menggunakan `cloud_id` dan otentikasi dasar dengan username

'elastic' dan password yang telah didefinisikan. Koneksi ke instans Elasticsearch diuji dengan metode `ping()`, dan pesan akan dicetak untuk menunjukkan apakah koneksi berhasil atau gagal. Skrip ini memastikan aplikasi Python dapat berinteraksi dengan Elasticsearch di cloud.

4.2.1 Import Library

```
pip install elasticsearch
pip install pyspark
pip install spotipy
pip install kibana-api
pip install pyelasticsearch
pip install Sastrawi
```

Perintah tersebut digunakan untuk menginstal berbagai pustaka Python yang mendukung analisis data dan pengelolaan aplikasi berbasis Elasticsearch. `elasticsearch` adalah pustaka untuk berinteraksi dengan instans Elasticsearch. `pyspark` memungkinkan penggunaan Apache Spark untuk analisis data skala besar. `spotipy` adalah klien Python untuk API Spotify, memudahkan akses dan manipulasi data dari Spotify. `kibana-api` menyediakan antarmuka untuk berinteraksi dengan Kibana, alat visualisasi data untuk Elasticsearch. `pyelasticsearch` adalah pustaka lain untuk berinteraksi dengan Elasticsearch, menawarkan fungsi tambahan. `Sastrawi` adalah pustaka untuk pemrosesan bahasa alami (NLP) khusus untuk bahasa Indonesia, mendukung tugas seperti stemming. Semua pustaka ini diinstal menggunakan `pip`, manajer paket untuk Python.

4.2.2 Import Modul

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import udf
from pyspark.sql.types import ArrayType, StringType
from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark.ml.feature import HashingTF, Tokenizer, StopWordsRemover, IDF
from pyspark.ml import Pipeline
from pyspark.ml.recommendation import ALS
from pyspark.ml.evaluation import RegressionEvaluator
import matplotlib.pyplot as plt

from pyspark.sql import SparkSession
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

Kode tersebut mengimpor berbagai modul dan fungsi dari pustaka `pyspark` untuk mendukung pemrosesan dan analisis data besar, serta pustaka lain untuk tugas NLP dan visualisasi. `SparkSession` adalah titik awal untuk menggunakan Spark SQL, sementara `udf` digunakan untuk mendefinisikan fungsi pengguna. `ArrayType`, `StringType`, dan

berbagai tipe data lainnya dari `pyspark.sql.types` membantu mendefinisikan skema data. Fungsi dari `pyspark.sql.functions` mendukung operasi transformasi data. Modul `pyspark.ml.feature` mengimpor alat untuk pemrosesan teks seperti `HashingTF`, `Tokenizer`, `StopWordsRemover`, dan `IDF`, serta `Pipeline` untuk mengelola alur kerja machine learning. `ALS` dan `RegressionEvaluator` digunakan untuk rekomendasi dan evaluasi model. `matplotlib.pyplot` diimpor untuk visualisasi. Selain itu, `re`, `word_tokenize`, `stopwords`, dan `Sastrawi` digunakan untuk tugas NLP, seperti tokenisasi dan stemming bahasa Indonesia.

4.2.3 Data Collection

```
import pandas as pd

data = pd.read_csv('final_topmusic_2023.csv', error_bad_lines=False)
data
```

Parameter `error_bad_lines=False` dalam perintah `pd.read_csv` digunakan untuk mengabaikan baris yang menyebabkan kesalahan saat membaca file CSV. Dengan pengaturan ini, setiap baris yang tidak dapat diparse dengan benar oleh `pandas` akan dilewati dan tidak dimasukkan ke dalam `DataFrame` yang dihasilkan, sehingga mencegah program berhenti atau gagal karena adanya baris yang rusak atau tidak sesuai format.

```
elastic.info()
```

```
ObjectApiResponse({'name': 'instance-0000000000', 'cluster_name': '5580afffdac54a8ba42c55573d560a0a', 'cluster_uuid': 'u5jjwYvJ  
S26ID0JXfoHUXg', 'version': {'number': '8.14.1', 'build_flavor': 'default', 'build_type': 'docker', 'build_hash': '93a57a1a76f5  
56d8aee6a90d1a95b06187501310', 'build_date': '2024-06-10T23:35:17.114581191Z', 'build_snapshot': False, 'lucene_version': '9.1  
0.0', 'minimum_wire_compatibility_version': '7.17.0', 'minimum_index_compatibility_version': '7.0.0'}, 'tagline': 'You Know, fo  
r Search'})
```

Kode ini digunakan untuk mendapatkan informasi dasar tentang kluster Elasticsearch yang terhubung. Ketika metode ini dipanggil, ia akan mengembalikan data seperti versi Elasticsearch yang berjalan, informasi tentang node, nama kluster, dan detail lainnya. Ini berguna untuk memastikan koneksi berhasil dan untuk memverifikasi berbagai atribut kluster sebelum melakukan operasi lebih lanjut.

4.2.4 Ubah DataFrame menjadi format JSON

```
import pandas as pd

# Assuming you have a Pandas DataFrame called 'df'
json_data = data.to_json(orient='records')

# Print the JSON data
print(json_data)
```

```
[{"name": "En Las Calles De Madrid", "album": "En Las Calles De Madrid", "artist": "EDM NATIVES", "release_date": "2023-10-25", "length": 209400.0, "popularity": 0.0, "danceability": 0.47, "acousticness": 0.0178, "energy": 0.97, "instrumentalness": 0.899, "liveness": 0.121, "loudness": -2.274, "speechiness": 0.0402, "tempo": 199.991, "time_signature": 4.0}, {"name": "BLACK JAM", "album": "BLACK JAM", "artist": "NEO JAPONISM", "release_date": "2023-02-15", "length": 194153.0, "popularity": 6.0, "danceability": 0.394, "acousticness": 0.00623, "energy": 0.954, "instrumentalness": 0.0, "liveness": 0.0572, "loudness": -2.083, "speechiness": 0.171, "tempo": 195.225, "time_signature": 4.0}, {"name": "Bludsteroo", "album": "Sugma, Vol. 4", "artist": "Friox", "release_date": "2023-11-19", "length": 161486.0, "popularity": 0.0, "danceability": 0.116, "acousticness": 0.116, "energy": 0.779, "instrumentalness": 0.643, "liveness": 0.121, "loudness": -9.112, "speechiness": 0.035, "tempo": 140.014, "time_signature": 4.0}, {"name": null, "album": null, "artist": null, "release_date": null, "length": null, "popularity": null, "danceability": null, "acousticness": null, "energy": null, "instrumentalness": null, "liveness": null, "loudness": null, "speechiness": null, "tempo": null, "time_signature": null}, {"name": "Durga Sapta Shloki", "album": "Sarveswari Durga Maa", "artist": "Various Artists", "release_date": "2023-11-17", "length": 172997.0, "popularity": 0.0, "danceability": 0.495, "acousticness": 0.785, "energy": 0.145, "instrumentalness": 0.0, "liveness": 0.111, "loudness": -16.433, "speechiness": 0.0608, "tempo": 170.412, "time_signature": 4.0}, {"name": "My Pronouns Are Usa", "album": "My Pronouns Are Usa", "artist": "Dubsokie", "release_date": "2023-08-18", "length": 195813.0, "popularity": 7.0, "danceability": 0.788, "acousticness": 0.0239, "energy": 0.829, "instrumentalness": 0.0, "liveness": 0.208, "loudness": -8.072, "speechiness": 0.096, "tempo": 129.964, "time_signature": 4.0}, {"name": "Cup Fest", "album": "Cup Fest", "artist": "SkillinJah", "release_date": "2023-09-07", "length": 149379.0, "popularity": 0.0, "danceability": 0.706, "acousticness": 0.00171, "energy": 0.806, "instrumentalness": 0.0, "liveness": 0.156, "loudness": -5.367, "speechiness": 0.375, "tempo": 145.129, "time_signature": 4.0}, {"name": "CAMARO", "album": "CAMARO", "artist": "DIDKER", "release_date": "2023-08-01", "length": 199480.0, "popularity": 0.0, "danceability": 0.631, "acousticness": 0.00557, "energy": 0.562, "instrumentalness": 0.854, "liveness": 0.115, "loudness": -13.688, "speechiness": 0.0465, "tempo": 153.996, "time_signature": 4.0}, {"name": "Soy un humano", "album": "Soy un humano", "artist": "Alonso", "release_date": "2023-11-17", "length": 289685.0, "popularity": 2.0, "danceability": 0.0, "acousticness": 0.0, "energy": 0.0, "instrumentalness": 0.0, "liveness": 0.0, "loudness": 0.0, "speechiness": 0.0, "tempo": 0.0, "time_signature": 0.0}]]
```

Kode tersebut mengimpor pustaka `pandas` dan mengonversi `DataFrame` data menjadi format JSON. Dengan menggunakan metode `to_json` dan parameter `orient='records'`, setiap baris dalam `DataFrame` diubah menjadi objek JSON terpisah dalam sebuah array. Hasil konversi ini kemudian disimpan dalam variabel `json_data`.

4.2.5 Mengindeks setiap data dari sebuah `DataFrame` ke dalam `Elasticsearch`

```
import json
from elasticsearch import Elasticsearch

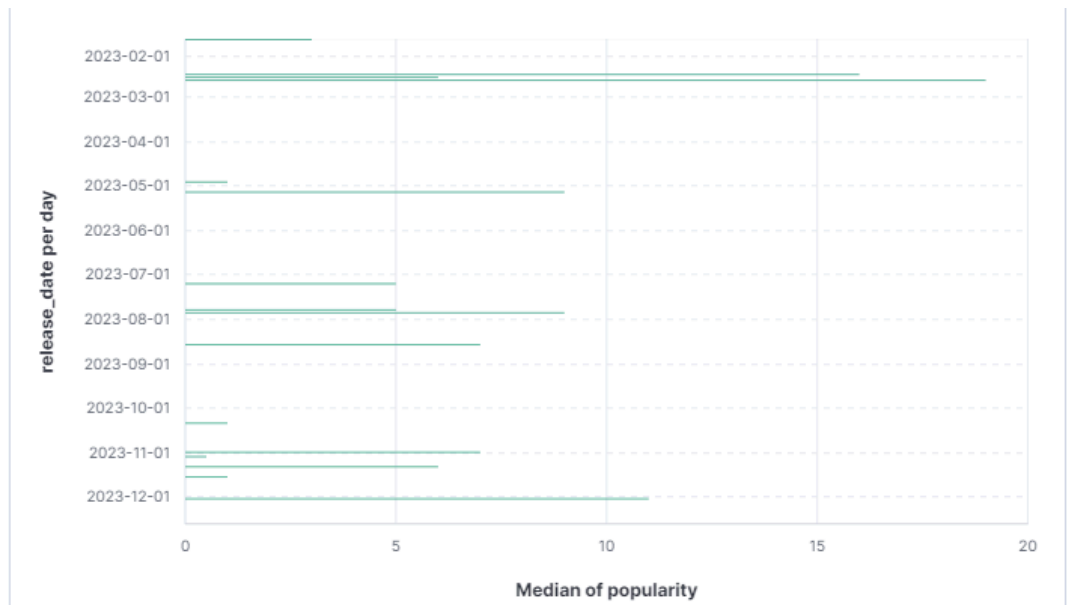
# Elasticsearch index
elastic_index = "spotify" # Replace with your desired index name

# Assuming you have a Pandas DataFrame called 'df'
json_data = data.to_json(orient='records')
rows = json.loads(json_data)
for row in rows:
    document_id = row["name"] # Replace with a unique identifier for each document
    elastic.index(index=elastic_index, id=document_id, body=row)

# Successful response!
print(elastic.info())

{'name': 'instance-0000000000', 'cluster_name': '5580afffdac54a8ba42c55573d560a0a', 'cluster_uuid': 'u5jjwYvJ526ID0JXfoHUXg', 'version': {'number': '8.14.1', 'build_flavor': 'default', 'build_type': 'docker', 'build_hash': '93a57a1a76f556d8aee6a90d1a95b06187501310', 'build_date': '2024-06-10T23:35:17.114581191Z', 'build_snapshot': False, 'lucene_version': '9.10.0', 'minimum_wire_compatibility_version': '7.17.0', 'minimum_index_compatibility_version': '7.0.0'}, 'tagline': 'You Know, for Search'}
```

Kode diatas mengunggah data dari `DataFrame` `Pandas` ke `Elasticsearch`. Pertama, data `DataFrame` dikonversi menjadi format JSON dan dimuat ke dalam daftar objek. Kemudian, setiap objek diunggah sebagai dokumen ke indeks `Elasticsearch` yang ditentukan menggunakan `elastic.index`. Nama indeks dan pengidentifikasi dokumen unik harus disesuaikan dengan kebutuhan spesifik. Terakhir, kode ini mencetak informasi tentang kluster `Elasticsearch` untuk memverifikasi koneksi dan status kluster.



Gambar 12. Visualiasai Release Songs

Dari grafik ini, terlihat bahwa pada bulan Februari dan Maret 2023 terdapat beberapa lagu yang memiliki median popularitas tinggi, menandakan bahwa lagu-lagu yang dirilis pada periode ini cukup populer di kalangan pendengar Spotify. Bulan Juli dan Agustus juga menunjukkan tren serupa, mungkin karena perilisan lagu-lagu musim panas yang sering kali mendapatkan perhatian lebih. Pada bulan November 2023, ada beberapa lagu dengan median popularitas yang cukup tinggi.

Sebaliknya, bulan-bulan seperti April, Mei, dan September menunjukkan sedikit atau hampir tidak ada lagu yang memiliki median popularitas tinggi, menandakan bahwa pada bulan-bulan tersebut tidak banyak lagu populer yang dirilis atau lagu-lagu yang dirilis tidak mendapatkan popularitas yang signifikan. Dengan menggunakan Elasticsearch, analisis ini memungkinkan untuk mendapatkan data secara real-time, memberikan wawasan yang cepat dan akurat mengenai tren populer lagu di Spotify sepanjang tahun. Grafik ini membantu dalam memahami kapan waktu-waktu tertentu dalam setahun di mana lagu-lagu yang dirilis mendapatkan popularitas tinggi, serta memberikan informasi yang berguna bagi industri musik dalam merencanakan perilisan lagu di masa depan.

4.2.6 Potongan script untuk Pemeriksaan Data Quality

```
import Extract
import pandas as pd

# Set of Data Quality Checks Needed to Perform Before Loading

def Data_Quality(load_df):
    #Checking Whether the DataFrame is empty
    if load_df.empty:
        print('No Songs Extracted')
        return False

    #Enforcing Primary keys since we don't need duplicates
    if pd.Series(load_df['played_at']).is_unique:
        pass
    else:
        #The Reason for using exception is to immediately terminate the
        program and avoid further processing raise
        Exception("Primary Key Exception,Data Might Contain
        duplicates")
```

Kode diatas merupakan bagian penting dari proses ETL (Extract, Transform, Load) yang bertujuan untuk menjaga kualitas dan integritas data sebelum dimuat ke dalam sistem lebih lanjut. Dengan memeriksa apakah DataFrame kosong dan memastikan tidak ada duplikat, kode ini membantu menghindari masalah yang mungkin terjadi di tahap pemrosesan data berikutnya.

BAB V

KESIMPULAN

Pada tahun 2023, lanskap musik digital di Spotify ditandai oleh popularitas tinggi beberapa lagu pada waktu tertentu. Penelitian menemukan bahwa bulan Februari dan Maret 2023 menampilkan beberapa lagu dengan median popularitas yang tinggi. Hal ini menunjukkan bahwa lagu-lagu yang dirilis pada periode ini mendapat sambutan hangat dari pendengar Spotify. Tren yang serupa terlihat pada bulan Juli dan Agustus, dipengaruhi oleh perilisan lagu-lagu musim panas yang mendapatkan perhatian lebih besar. Di sisi lain, bulan-bulan seperti April, Mei, dan September menunjukkan sedikit atau bahkan tidak ada lagu yang mencatat median popularitas tinggi, menunjukkan bahwa pada periode-periode tersebut popularitas lagu-lagu tidak terlalu signifikan atau mungkin sedang dalam fase kurang diminati oleh pendengar. Dari analisis ini, dapat dipahami bahwa popularitas lagu di Spotify pada tahun 2023 sangat dipengaruhi oleh waktu perilisan, yang mempengaruhi keterlibatan dan preferensi pendengar secara signifikan sepanjang tahun. Analisis ini memberikan wawasan mendalam tentang pola perilisan lagu dan dinamika popularitasnya dalam konteks platform streaming musik Spotify.

Dengan menggunakan Elasticsearch, penulis dapat menganalisis data real-time dari Spotify dan mengidentifikasi tren lagu populer dengan cepat dan akurat. Elasticsearch juga memungkinkan untuk mengidentifikasi faktor-faktor yang mempengaruhi popularitas sebuah lagu, seperti karakteristik musik, genre, dan artis. Dengan demikian, penelitian ini sudah mencapai tujuan yang kami inginkan, yaitu menganalisis tren populer lagu Spotify pada tahun 2023 dengan menggunakan Elasticsearch untuk analisis data real-time lagu dari Spotify. Hasil analisis ini dapat memberikan wawasan yang berguna bagi industri musik, termasuk artis, label rekaman, dan manajer, dalam memahami tren lagu populer dan mengembangkan strategi pemasaran yang lebih efektif.

DAFTAR PUSTAKA

- Elasticsearch: Apa Artinya, Dan Mengapa Anda Membutuhkannya? (2020-2024) <https://ichi.pro/id/elasticsearch-apa-artinya-dan-mengapa-anda-membutuhkannya-71597910397895>
- Alamsyah Putra, Aurelie, A. C., Khairunnisa Junaidi, & Siskia Israwana. (2023). *Implementasi MapReduce Pada Dataset Spotify Top Music Untuk Mengetahui Artist yang paling banyak Didengar Dalam Kurun Waktu 10 Tahun*. <https://jurnal.netplg.com/index.php/jnca/article/view/79>
- Anggraini, N., Okananta, R., Musaffak, A. L., Salsabila, N., Balqhis, A., & Belardo, V. (2023). *Analisis Real-Time Data Youtube Dengan PySpark, Elasticsearch, dan Kibana*.
- Bunga DEa Laraswati. (2023). Apa itu Analisis Data Real-time dan Mengapa Penting? <https://blog.algorit.ma/analisis-data-real-time/>
- Contact. (n.d.). Spotify. Retrieved June 20, 2024, from <https://www.spotify.com/us/about-us/contact/>
- Dylan Crawshaw. (2020). Spotify's Web API Explained. <https://dcrawsh.medium.com/spotify-s-web-api-explained-7d0bea7dcfab>
- Immanulloh, D. T., Hariyanto, V., Oktaviana, E., & Kamilah, H. (2024). *PREDIKSI TREN MUSIK TAHUN 2024 PADA PLATFORM SPOTIFY MENGGUNAKAN METODE RANDOM FOREST REGRESSOR Music Trend Prediction In 2024 On The Spotify Platform Using The Random Forest Regressor Method*. https://www.researchgate.net/publication/376397973_PREDIKSI_TREN_MUSIK_TAHUN_2024_PADA_PLATFORM_SPOTIFY_MENGGUNAKAN_METODE_RANDOM_FOREST_REGRESSOR_Music_Trend_Prediction_In_2024_On_The_Spotify_Platform_Using_The_Random_Forest_Regressor_Method
- McKinsey. (2011). *Big data: The next frontier for innovation, competition, and productivity*. https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI_big_data_full_report.pdf
- Nuriska, D., Irawan, B., Bachtiar, A., & Dikananda, A. R. (2023). *KLASTERISASI DATA LAGU TERPOPULER SPOTIFY 2023 BERDASARKAN SUASANA HATI MENGGUNAKAN ALGORITMA K-MEANS*. <https://ejournal.itn.ac.id/index.php/jati/article/view/8232>
- Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya. (2023, May 31). Dicoding. Retrieved June 20, 2024, from <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>
- Sartika, E. P., & Cahyono, A. B. (2020). *PEMANFAATAN ELASTICSEARCH LOGSTASH KIBANA UNTUK PENGELOLAAN DAN VISUALISASI DATA PORTAL PENGEMBANGAN DAN PEMBINAAN SUMBER DAYA MANUSIA (PORTAL PPSDM)*. <https://dspace.uui.ac.id/handle/123456789/20219?show=full>
- spotify.com. (2024). Apa itu Spotify? <https://support.spotify.com>
- Web API. (2024). <https://developer.spotify.com/documentation/web-api>
- elastic.co. (2024). *The Elastic Search AI Platform*. <http://www.elastic.co>

LAMPIRAN

1. Data hasil scraping secara real-time Spotify 2023

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	name	album	artist	release_date	length	popularity	danceabil	acousticn	energy	instrumer	liveness	loudness	speechinc	tempo	time_signature	
2	En Las Cal	En Las Cal	EDM NATI	10/25/2023	209400	0	0.47	0.0178	0.97	0.899	0.121	-2.274	0.0402	199.991	4	
3	BLACK JAN	BLACK JAN	NEO JAPO	2/15/2023	194153	6	0.394	0.00623	0.954	0	0.0572	-2.083	0.171	195.225	4	
4	Violin Cor	RV 229: III	Vivaldi: C	Antonio Vivaldi	9/22/2023	198802	5	0.402	0.916	0.171	0.259	0.118	-19.284	0.0669	68.653	4
5	Bludstereo	"Sugma, Vol. 4"	Friox,2023-11-19,	161486,0,0.716,0.116,0.779,0.643,0.121,-9.112,0.035,140.014,4												
6	Durga Sap	Sarveswat	Various Ai	11/17/2023	172997	0	0.495	0.785	0.145	0	0.111	-16.433	0.0608	170.412	4	
7	My Pronoi	My Pronoi	Dubskie	8/18/2023	195813	7	0.788	0.0239	0.829	0	0.208	-8.072	0.096	129.964	4	
8	Cup Fest	Cup Fest	SkillinJah	9/7/2023	149379	0	0.706	0.00171	0.806	0	0.156	-5.367	0.375	145.129	4	
9	CAMARO	CAMARO	DIDKER	8/1/2023	199480	0	0.631	0.00557	0.562	0.854	0.115	-13.688	0.0465	153.996	4	
10	Soy un hu	Soy un hu	Alonso	11/17/2023	289685	2	0.615	0.392	0.603	0.778	0.13	-7.908	0.0273	125.392	4	
11	Never Let	Never Let	Diskatch	11/10/2023	176008	6	0.726	0.0394	0.642	0.358	0.109	-10.427	0.0502	124.011	4	
12	Patience	Patience	Quinn Ay	4/28/2023	120000	1	0.629	0.814	0.57	1.95E-06	0.156	-9.332	0.464	163.981	4	
13	Bir ÄziÄ	Ye Klasikler	MÄÄsiÄÄ	2/10/2023	233174	0	0.348	0.197	0.682	0	0.444	-6.269	0.052	132.226	4	
14	Your Eyes	Your Eyes	sxhinoph	2/17/2023	146827	9	0.459	0.00338	0.828	0.116	0.136	-9.886	0.0361	169.984	4	
15	Muevelo,	"Christmas	Dance Party	2023-2024 (Best of Dance, House & Electro)",	Various Artists,	2023-12-22,	147554,0,0.805,0.0328,0.936,0.771,0.0749,-3.321,0.0474,117.048,4									
16	Close Da	C Close Da	BlakKyd	8/12/2023	108017	0	0.783	0.00444	0.59	0	0.0894	-9.083	0.331	100.167	4	
17	Ballad of t	The Leger	Flying Puf	2/13/2023	51822	16	0.739	0.987	0.327	0.799	0.171	-14.04	0.077	94.988	4	
18	Quantum	Synthetic	CHINCON	10/2/2023	64000	0	0.518	0.287	0.233	0.959	0.113	-19.983	0.0361	74.924	4	
19	Nice Weal	Nice Weal	LoFi Robo	11/11/2023	174000	0	0.663	0.458	0.245	0.893	0.108	-21.269	0.0607	79.987	4	
20	Rubi Rubi	Back To Th	Cherry Bai	11/3/2023	188983	0	0.597	0.225	0.718	0.00025	0.0464	-8.346	0.0326	101.076	4	
21	Awake - Ir	INSTRUM	ÄDV!DE	12/15/2023	171212	0	0.445	0.37	0.392	0.855	0.104	-15.112	0.0635	78.949	4	
22	Stajt Tkun	Stajt Tkun	Sean Borg	7/7/2023	170438	1	0.611	0.686	0.325	9.94E-06	0.196	-11.033	0.0257	85.013	4	
23	Faux Bam!	How To Re	dEUS	2/17/2023	267760	19	0.534	0.0311	0.677	0.000613	0.313	-8.475	0.0357	145.974	4	
24	Right Here	Right Here	YUNIQUE.	7/21/2023	133333	0	0.71	0.496	0.741	0.119	0.103	-8.129	0.123	126.108	4	
25	far away	twilight	LUKEBIKE	9/20/2023	135725	0	0.556	0.0109	0.631	0.0249	0.245	-7.929	0.0326	87.311	5	
26	PrÄÄs	PrÄÄs	Karamboli	9/15/2023	594471	0	0.372	0.0527	0.521	0.801	0.356	-10.113	0.0336	128.362	4	
27	Savage	Savage	C. M. G.	9/30/2023	78628	0	0.463	0.00223	0.624	0.858	0.139	-9.079	0.196	148.209	4	

2. Presentasi Project bab 1 dan bab 2 :

[Presentasi Bab 1-2](#)