**BINAR CHALLENGE GOLD**

**DATA SCIENCE 25**

**“PEMBUATAN A.P.I UNTUK MEMPROSESS DATA DAN EXPLORE DATA ANALYSIS”**

A logo with a purple flower

Description automatically generated

**DISUSUN OLEH:**

**DEVAN ALINGGA ABDA**

**DSC240800013**

# **Pendahuluan**

## **Latar Belakang Masalah**

Pada zaman sekarang, media sosial dan platform online lainnya dipenuhi dengan penggunaan bahasa informal, termasuk penggunaan kata-kata slang (alay) dan kata-kata yang bernada kasar. Hal ini bisa menjadi tantangan tersendiri ketika teks tersebut perlu diproses secara lebih lanjut, baik itu untuk analisis teks, pembuatan laporan, atau digunakan dalam aplikasi berbasis bahasa yang lebih formal. Oleh karena itu, konversi kata-kata ini menjadi bentuk yang lebih normal atau formal sangat diperlukan.

Dalam pengolahan data teks di era digital saat ini, seringkali ditemui kata-kata yang tidak formal, atau yang lebih dikenal dengan istilah **"kata alay"** dan juga kata-kata **abusive**. Kata-kata alay merupakan gaya bahasa yang cenderung menggunakan singkatan-singkatan atau ejaan yang tidak sesuai dengan kaidah bahasa formal, sementara kata-kata abusive mengandung unsur kebencian atau penghinaan yang berpotensi merusak validitas dan etika sebuah data. Di dunia akademik maupun industri, ketika data yang mengandung unsur alay dan abusive ini ingin diolah lebih lanjut, terutama untuk kebutuhan pengembangan model berbasis **Natural Language Processing (NLP)**, data tersebut harus dibersihkan terlebih dahulu agar dapat diinterpretasi dengan baik oleh **API** yang bertanggung jawab dalam pemrosesan data tersebut.

Tujuan utama dari proyek ini adalah melakukan transformasi atau konversi kata-kata alay menjadi kata formal dan menghapus kata-kata abusive dari dataset. Proses ini penting karena banyak algoritma pengolahan teks modern yang mengandalkan kebersihan dan formalitas data agar dapat memberikan hasil yang akurat. Dalam proyek ini, digunakan beberapa file pendukung seperti new\_kamusalay.csv yang berisi kamus konversi dari kata-kata alay menjadi kata formal, serta abusive.csv yang memuat daftar kata-kata abusive yang perlu dihilangkan dari data. Selain itu, file data.csv digunakan sebagai dataset yang akan diproses untuk kemudian dibersihkan dari kata-kata tidak layak tersebut.

Dengan memproses dataset menjadi lebih bersih dan formal, data menjadi lebih mudah untuk diproses oleh API dan berbagai alat analitik lainnya. Selain itu, dengan penghapusan kata abusive, kita dapat meminimalisir munculnya bias atau hasil analisis yang tidak etis. Hal ini penting baik dari sisi akademik maupun bisnis, karena data yang bersih tidak hanya meningkatkan performa model pembelajaran mesin, tetapi juga menjaga integritas dan akurasi hasil analisis yang dihasilkan.

## **Rumusan Masalah**

1. Bagaimana cara membersihkan dataset teks yang mengandung kata-kata alay sehingga dapat diubah menjadi kata-kata formal tanpa mengurangi makna asli dari teks tersebut?
2. Bagaimana cara mendeteksi dan menghapus kata-kata abusive dalam dataset agar hasil analisis lebih valid, etis, dan sesuai dengan standar bahasa yang formal?
3. Bagaimana mekanisme penyimpanan dan pengelolaan hasil pemrosesan teks (konversi kata alay dan penghapusan kata abusive) ke dalam database SQLite untuk memudahkan akses dan penggunaan data lebih lanjut?

## **Tujuan Penelitian**

1. Mengembangkan metode untuk mengubah kata-kata alay dalam dataset teks menjadi kata-kata formal agar data lebih mudah diproses oleh sistem analitik dan API.
2. Merancang sistem untuk mendeteksi dan menghapus kata-kata abusive dalam dataset teks guna meningkatkan kualitas, validitas, dan etika data yang diproses.
3. Membangun mekanisme penyimpanan data hasil pemrosesan teks ke dalam database SQLite yang memungkinkan penyimpanan dan pengambilan data yang efisien.

# **Metode Penelitian**

## **Deskripsi Data**

1. “data.csv”A black table with white text

   Description automatically generated

Deskripsi: File ini berisi data mentah (raw data) yang akan diproses. Biasanya, file ini berisi teks yang mungkin mengandung kata-kata alay dan abusive. Kolom dalam file ini kemungkinan berisi satu atau lebih kolom teks yang belum dibersihkan, yang memerlukan transformasi menjadi lebih formal dan bersih dari kata-kata yang tidak pantas.

Isi: Kolom teks yang belum diolah, berpotensi mengandung kata-kata slang, alay, atau abusive dan data ini akan menjadi fokus utama dalam proses pembersihan dan konversi.

1. “abusive.csv”

A black rectangular object with white text

Description automatically generated

Deskripsi: File ini memuat daftar kata-kata yang dianggap abusive atau tidak pantas. Kata-kata dalam file ini harus dihapus dari dataset untuk menjaga integritas dan etika data. Kata-kata yang termasuk dalam kategori abusive umumnya mengandung unsur penghinaan, kebencian, atau hal-hal yang tidak etis dan sensitif.

Isi: Satu kolom yang berisi kata-kata abusive dalam bahasa Indonesia. Setiap baris merepresentasikan satu kata yang dianggap tidak pantas.

1. “kamus\_alay.csv”

A screenshot of a computer

Description automatically generated

Deskripsi: File ini adalah kamus konversi yang berisi daftar kata alay (slang) beserta padanan formalnya. File ini digunakan untuk mentransformasikan teks dalam dataset menjadi lebih formal sehingga lebih mudah dipahami oleh sistem atau model analitik. Kata alay di sini mencakup bahasa informal atau ejaan yang tidak baku yang sering digunakan di media sosial atau pesan singkat.

Isi: Dua kolom: Kolom 1: Kata alay (slang), Kolom 2: Kata formal yang merupakan padanan dari kata alay tersebut.

## **Metode Analisa Data**

Penelitian ini berfokus pada proses pembersihan data teks yang mengandung kata-kata alay dan abusive, agar dataset dapat digunakan untuk proses analisis lebih lanjut dengan API yang membutuhkan data bersih dan formal. Berikut adalah langkah-langkah metode penelitian yang dilakukan:

### **1. Proses Code Function Cleansing**

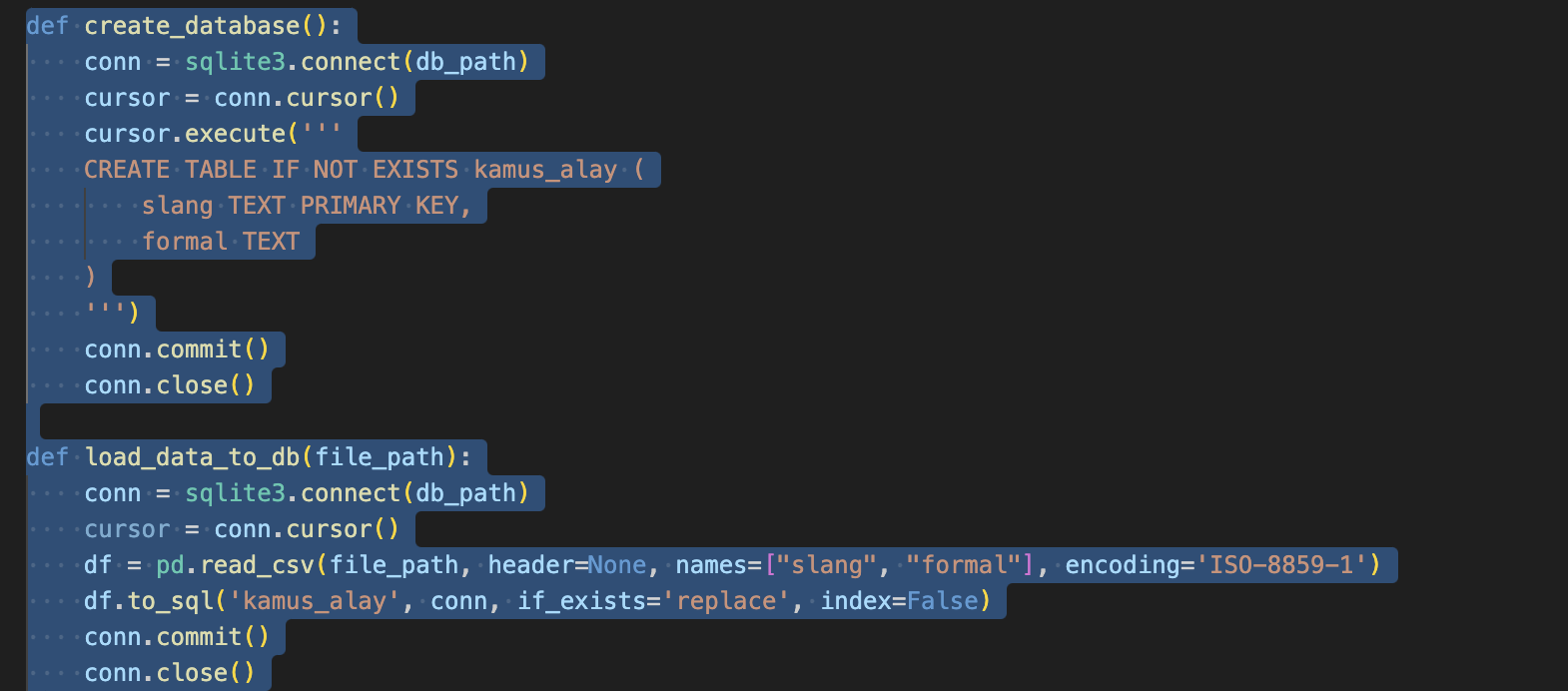
1. Proses Import Libraries

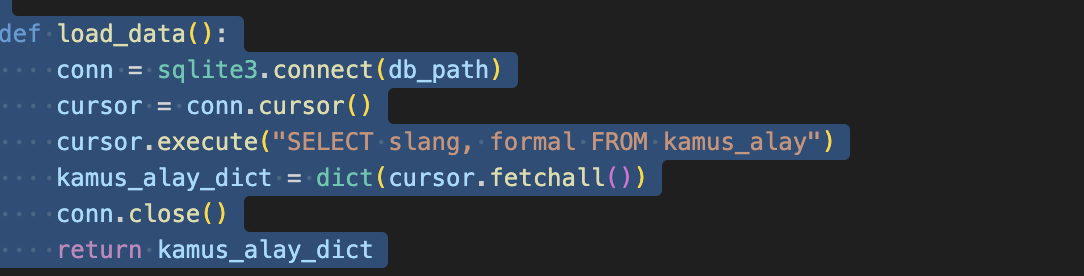


Bagian import digunakan untuk memasukkan modul atau pustaka eksternal yang diperlukan dalam skrip Python Anda. Modul ini menyediakan fungsi-fungsi tambahan yang tidak tersedia secara default di Python.

* gradio: Digunakan untuk membangun antarmuka pengguna yang memungkinkan interaksi dengan model atau aplikasi data.
* pandas: Mempermudah pengolahan data seperti membaca, membersihkan, dan menganalisis dataset.
* sqlite3: Memungkinkan skrip Python untuk membuat, mengakses, dan memodifikasi database SQLite.
* file\_path = './new\_kamusalay.csv': Menunjukkan lokasi file CSV yang berisi data slang dan formal yang akan diproses.
* data\_path = './archive/data.csv': Menunjukkan lokasi file CSV yang berisi dataset tambahan di dalam folder archive.
* abusive\_path = './archive/abusive.csv': Menunjukkan lokasi file CSV berisi data kata-kata kasar yang mungkin akan diproses atau di-filter dari dataset utama.

1. Membuat Database Sqlite 3





Penjelasan Fungsi:

• \*\*`create\_database()`\*\*: Membuat database SQLite dan tabel `kamus\_alay` jika belum ada, yang berisi dua kolom: `slang` dan `formal`. Fungsi ini memastikan database siap digunakan.

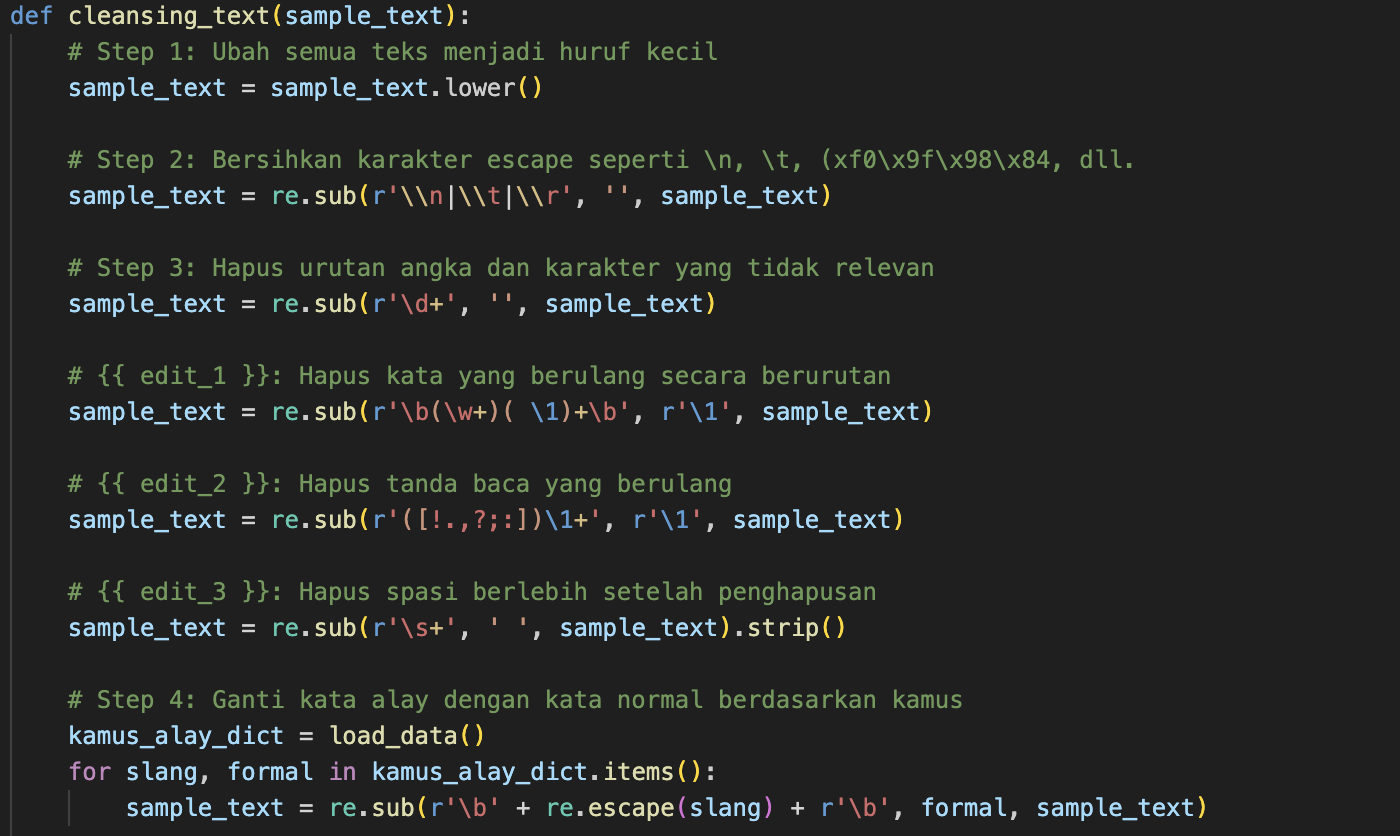
• \*\*`load\_data\_to\_db(file\_path)`\*\*: Membaca file CSV yang berisi pasangan kata slang dan formal, kemudian memasukkan data tersebut ke dalam tabel `kamus\_alay` di database SQLite. Jika tabel sudah ada, isinya akan digantikan.

• \*\*`load\_data()`\*\*: Mengambil semua data slang dan formal dari tabel `kamus\_alay` di database dan mengembalikannya sebagai kamus (\*dictionary\*) untuk penggunaan selanjutnya.

1. Process Cleansing

Proses cleansing atau pembersihan teks bertujuan untuk mengonversi kata-kata alay menjadi kata formal serta menghapus kata-kata abusive dari dataset. Fungsi cleansing yang digunakan dalam penelitian ini mencakup beberapa langkah utama:

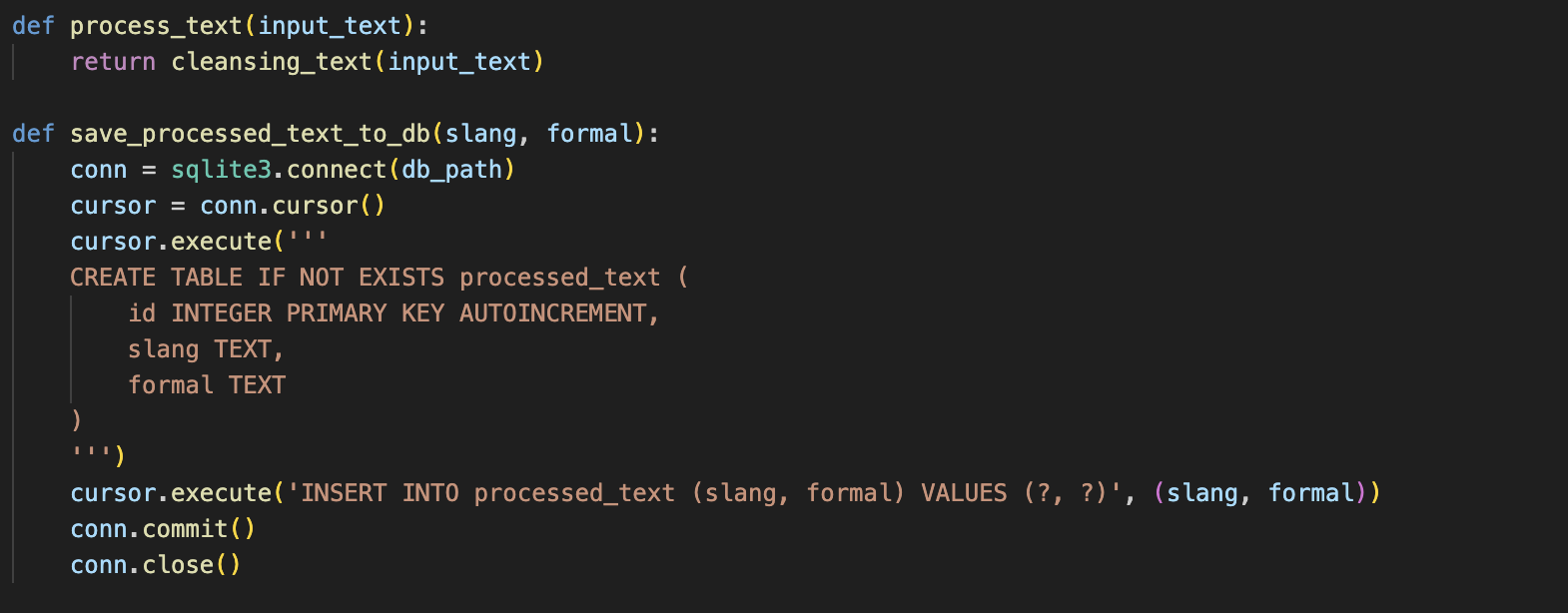
* **Lowercasing**: Mengubah semua teks menjadi huruf kecil untuk memastikan keseragaman.
* **Pembersihan Karakter Tidak Relevan**: Menghilangkan karakter-karakter tidak relevan seperti angka, tanda baca berlebih, serta karakter-karakter khusus.
* **Penggantian Kata Alay dengan Kata Formal**: Menggunakan kamus alay yang sudah diinput ke dalam SQLite untuk melakukan konversi kata-kata alay menjadi kata formal. Fungsi ini menggunakan regex untuk mencocokkan kata alay di teks dengan yang ada di kamus.
* **Penghapusan Kata Abusive**: Setelah kata alay dikonversi, langkah berikutnya adalah menghapus kata-kata abusive yang ditemukan di teks berdasarkan daftar kata abusive yang disediakan dalam file *abusive.csv*.



A screen shot of a computer program

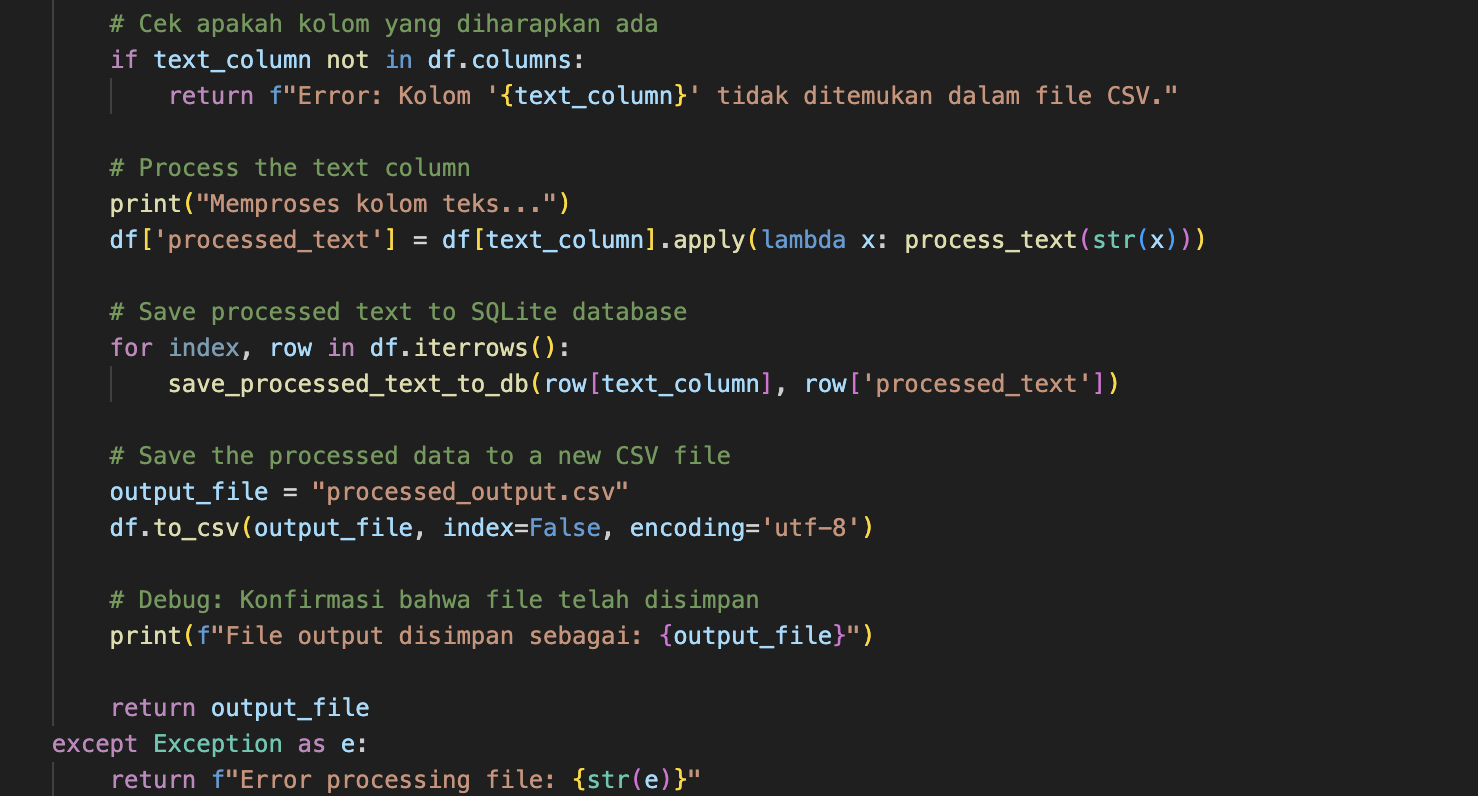
Description automatically generated

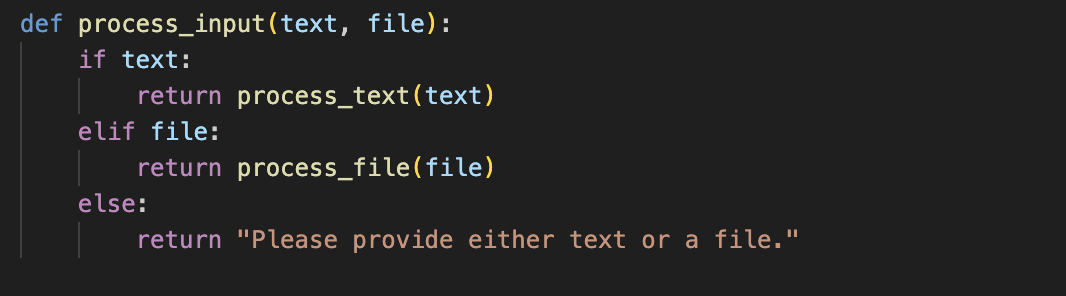
1. Pembuatan Gradio API



A screen shot of a computer program

Description automatically generated





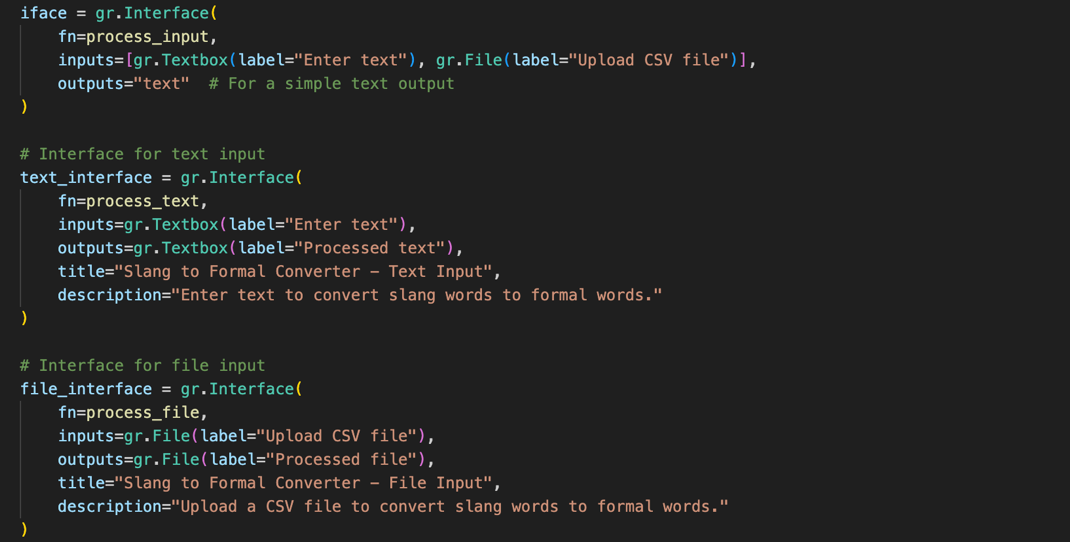
• process\_text(input\_text): Memproses teks masukan menggunakan fungsi cleansing\_text() untuk membersihkan teks dari kata-kata slang atau tidak formal.

• save\_processed\_text\_to\_db(slang, formal): Menyimpan pasangan slang dan teks formal yang sudah diproses ke tabel processed\_text di database SQLite. Jika tabel belum ada, maka tabel akan dibuat terlebih dahulu.

• process\_file(file): Membaca file CSV yang diunggah, memproses kolom teks pertama untuk menggantikan slang dengan teks formal, dan menyimpan hasilnya ke database serta menyimpannya dalam file CSV baru bernama processed\_output.csv. Juga mencakup pengecekan kesalahan seperti file kosong atau kolom yang hilang.

• process\_input(text, file): Memproses input berupa teks atau file, dan mengembalikan hasil pembersihan. Jika pengguna memberikan teks, teks akan diproses; jika file diunggah, file tersebut akan diproses.

1. Pembuatan Interface Gradio



A computer screen with text

Description automatically generated

• iface = gr.Interface(...): Membuat antarmuka Gradio untuk fungsi process\_input, menerima teks atau file CSV sebagai input dan mengeluarkan teks yang telah diproses.

• text\_interface = gr.Interface(...): Antarmuka untuk input teks, menggunakan fungsi process\_text, yang menampilkan teks yang telah diproses dalam textbox output.

• file\_interface = gr.Interface(...): Antarmuka untuk mengunggah file CSV, menggunakan fungsi process\_file, yang menyimpan hasil pemrosesan dalam file CSV baru.

• gr.TabbedInterface([...]): Menggabungkan kedua antarmuka (teks dan file) dalam satu tab untuk kemudahan akses.

• create\_database(): Memastikan bahwa database SQLite telah dibuat sebelum memproses data.

• load\_data\_to\_db(file\_path): Memuat data dari file CSV ke dalam database agar semua slang dan kata formal tersedia untuk pemrosesan.

1. Tampilan API

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### **2. Pemilihan Metode Statistik dan EDA (Exploratory Data Analysis)**

Metode ini digunakan untuk memahami karakteristik atau pola dasar dari data. Dalam konteks ini, Descriptive Analytics digunakan untuk memberikan gambaran mengenai kata-kata alay dan abusive yang ditemukan dalam dataset, serta jumlah dan jenis kata yang berhasil diubah atau dihapus setelah proses cleansing. Pengaplikasian dalam hal ini analisis deskriptif dapat digunakan untuk menghitung jumlah kata alay yang diubah menjadi formal dan seberapa banyak kata abusive yang berhasil dihapus dari dataset. Ini memberikan gambaran awal mengenai kualitas data yang diolah.

Dalam EDA, kita akan memvisualisasikan frekuensi kata alay dan abusive dalam dataset sebelum dan sesudah pembersihan. Metode ini membantu memberikan insight apakah pembersihan yang dilakukan berhasil mengubah karakteristik data.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Ketiga DataFrame yang ditampilkan terkait dengan analisis data tweet. abusive\_df berisi informasi tentang teks tweet beserta indikator biner yang menunjukkan apakah tweet tersebut mengandung ujaran kebencian (HS) atau bahasa kasar (Abusive), serta jenis target dari ujaran kebencian, seperti individu, kelompok, agama, atau ras. data\_df memiliki struktur yang serupa dengan abusive\_df, yang juga berfokus pada klasifikasi tweet berdasarkan ujaran kebencian dan bahasa kasar, namun mungkin berfungsi sebagai dataset utama atau pelengkap. Sementara itu, new\_kamusalay\_df berisi istilah slang atau bahasa gaul, khususnya yang terkait dengan dialek atau penggunaan bahasa informal di Jakarta, dengan terjemahan atau arti yang sesuai. Dataset ini berguna untuk memahami nuansa bahasa sehari-hari dalam konteks media sosial, terutama untuk analisis sentimen atau monitoring sosial. Secara keseluruhan, ketiga DataFrame ini mendukung analisis mengenai pola ujaran kebencian, bahasa kasar, dan penggunaan bahasa sehari-hari dalam tweet di Indonesia.

1. Langkah 2 : Membaca Nilai paling akhir dari data

A screenshot of a computer

Description automatically generatedA screenshot of a computer

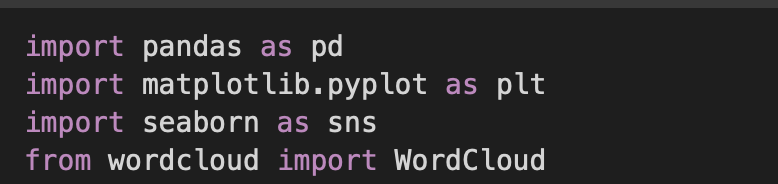
Description automatically generatedA screenshot of a computer

Description automatically generated

abusive\_df dan data\_df menampilkan bagian akhir dari data yang menunjukkan tweet dengan indikator kebencian dan bahasa kasar. Setiap baris berisi tweet, dengan kolom HS yang menandakan adanya ujaran kebencian, dan Abusive untuk mendeteksi bahasa kasar. Kolom seperti HS\_Individual dan HS\_Group membantu mengidentifikasi target kebencian, baik individu maupun kelompok, serta kategori lain seperti HS\_Religion dan HS\_Race untuk mendeteksi kebencian berdasarkan agama dan ras.

new\_kamusalay\_df menampilkan akhir dari kamus kata-kata gaul atau singkatan yang digunakan dalam bahasa sehari-hari, khususnya dalam konteks informal di Indonesia. Contohnya, "mendikbud" diterjemahkan menjadi "menteri pendidikan dan kebudayaan" dan "mendag" menjadi "menteri perdagangan". Kamus ini membantu dalam memahami bahasa gaul atau singkatan yang mungkin muncul dalam tweet, memudahkan analisis teks dengan istilah lokal.

1. Langkah 3 : Membaca Library yang mendukung EDA

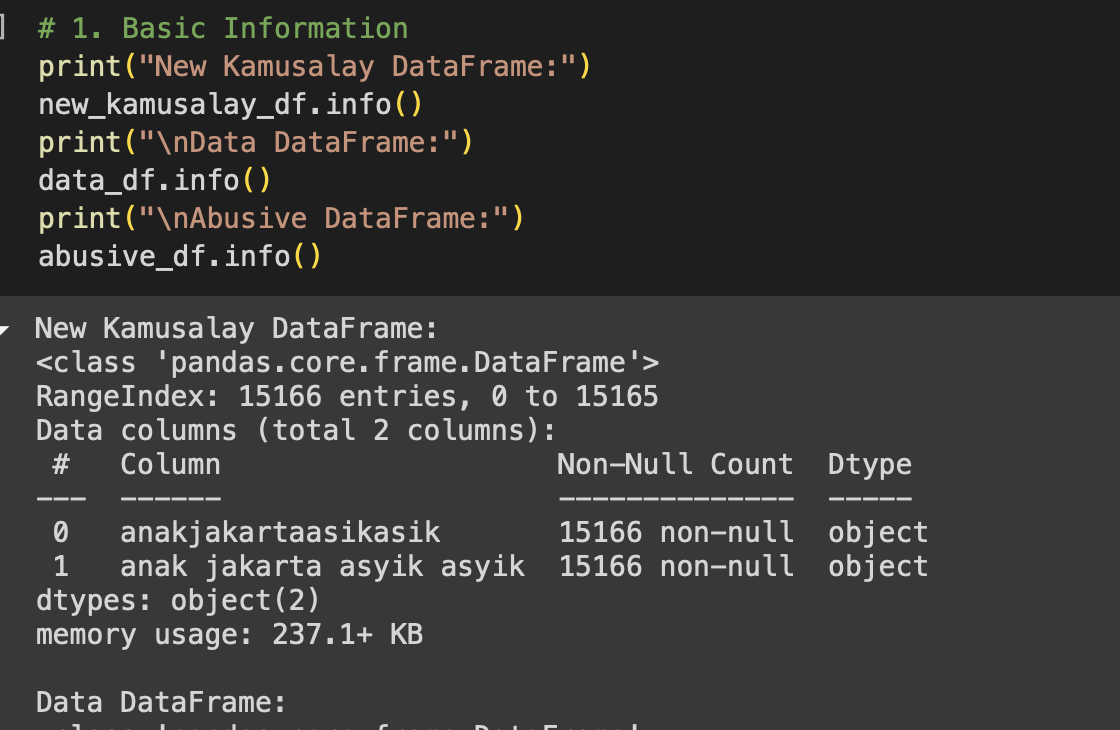


pandas: Untuk manipulasi data.

matplotlib.pyplot dan seaborn: Untuk membuat visualisasi data.

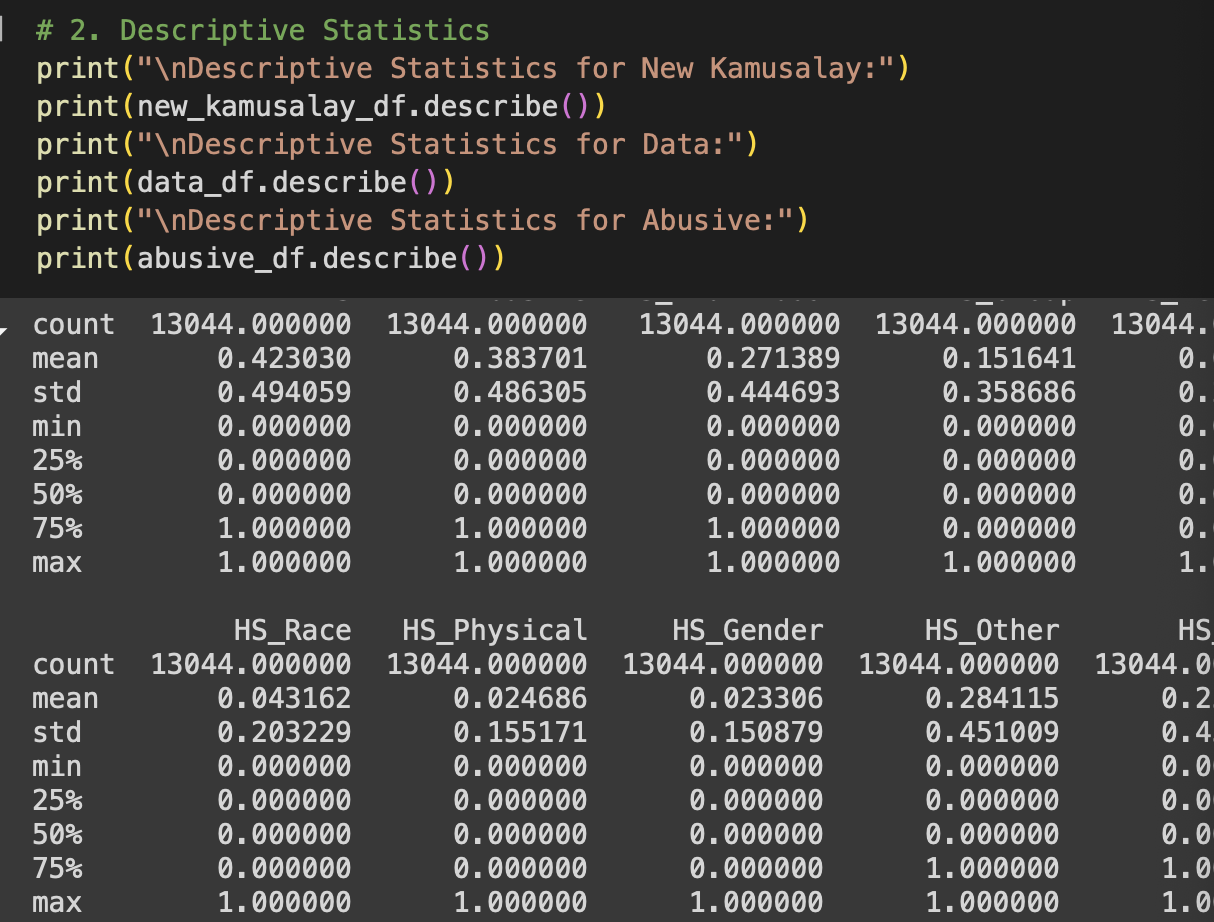
WordCloud: Untuk membuat visualisasi berbentuk awan kata.

1. Langkah 4 : Membaca Basic Information



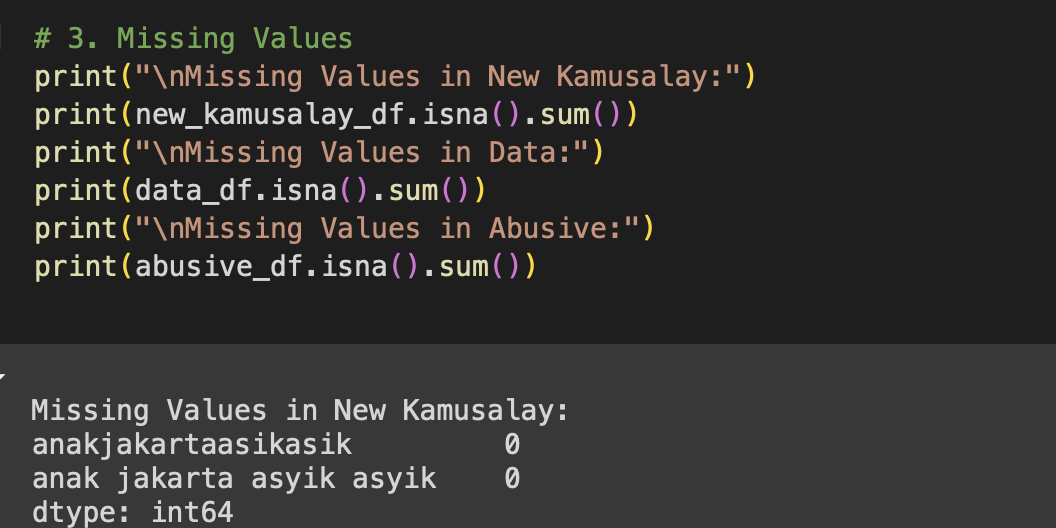
Kode di bawahnya menampilkan informasi dasar dari setiap DataFrame dengan menggunakan metode info(). Fungsi ini memberikan gambaran umum tentang jumlah baris, tipe data, dan jumlah nilai non-null di setiap kolom dari new\_kamusalay\_df, data\_df, dan abusive\_df. Langkah ini penting dalam Exploratory Data Analysis (EDA) untuk memahami struktur dan kelengkapan data sebelum melanjutkan ke analisis lebih lanjut atau visualisasi.

1. Langkah 5: Mencari Nilai Statiska Deskriptifnya



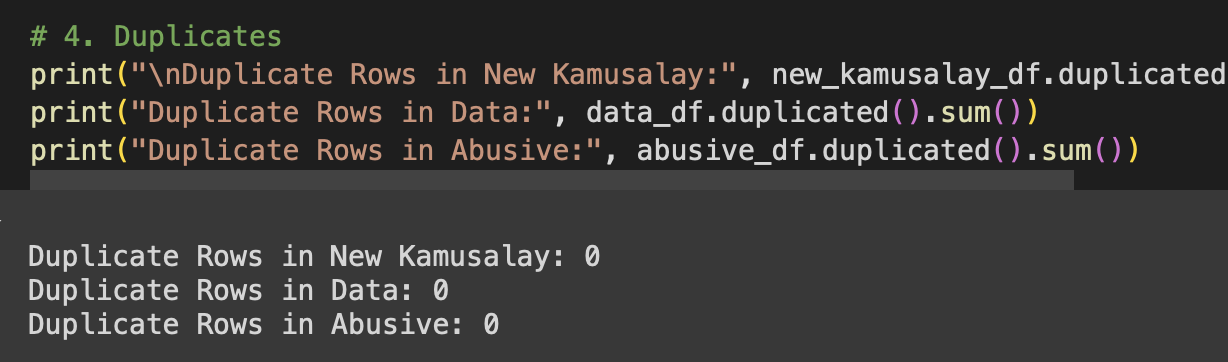
Statistik Deskriptif: Fungsi describe() digunakan untuk mendapatkan statistik deskriptif dasar dari setiap DataFrame. Untuk new\_kamusalay\_df, terlihat bahwa terdapat 15.166 baris unik dalam kolom pertama, dengan nilai paling sering muncul adalah "pakcikdahtua" yang hanya muncul sekali. Untuk data\_df dan abusive\_df, fungsi describe() memberikan ringkasan statistik untuk kolom numerik, termasuk jumlah (count), nilai rata-rata (mean), standar deviasi (std), serta nilai minimum dan maksimum untuk masing-masing kolom numerik.

1. Langkah 6 : Mencari Nilai Missing Valusnya



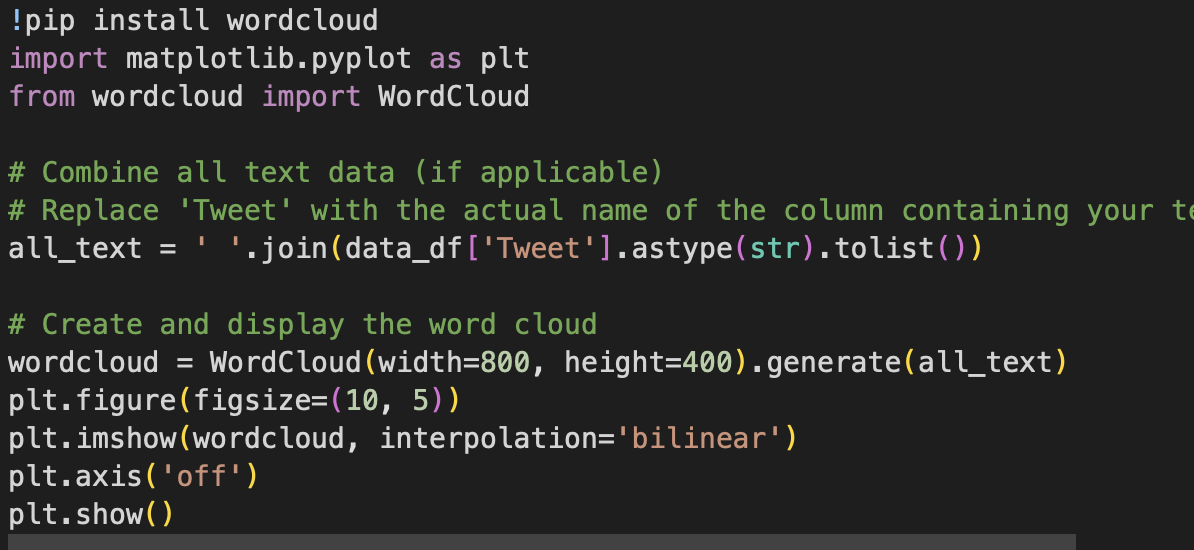
Pemeriksaan Missing Values: Kode ini menggunakan fungsi isna() dan sum() untuk menghitung jumlah nilai yang hilang (missing values) di setiap kolom dari ketiga DataFrame. Hasilnya menunjukkan bahwa semua kolom di new\_kamusalay\_df, data\_df, dan abusive\_df tidak memiliki nilai yang hilang (nilai 0 untuk setiap kolom). Data tanpa missing values memudahkan analisis lebih lanjut karena tidak perlu melakukan imputasi atau pembersihan tambahan.

1. Langkah 7 : Mencari Nilai Duplikat



Deteksi Duplikat: Kode ini menggunakan fungsi duplicated() untuk memeriksa baris yang duplikat pada masing-masing DataFrame: new\_kamusalay\_df, data\_df, dan abusive\_df. Hasil sum() dari fungsi tersebut menunjukkan bahwa tidak ada baris duplikat di ketiga DataFrame (nilai 0 untuk masing-masing). Ini penting untuk memastikan bahwa data yang dianalisis tidak mengandung entri ganda, yang bisa memengaruhi hasil analisis.

1. Langkah 8 : Mencari Kata Kata yang paling banyak digunakan

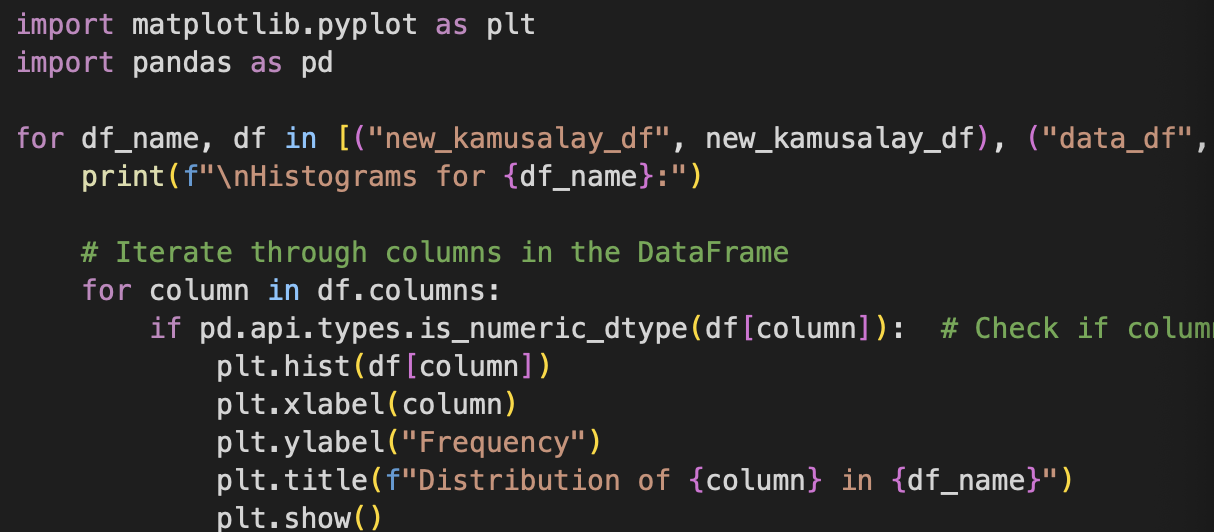


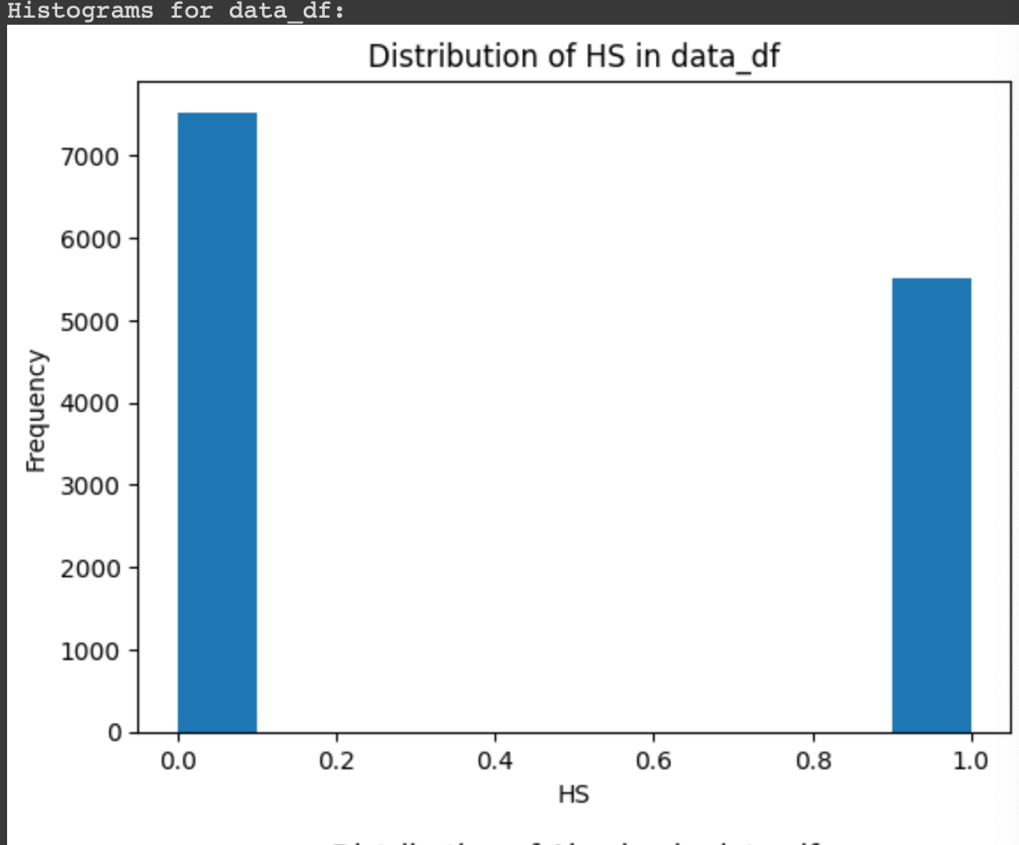


Word Cloud Code: Langkah-langkahnya sebagai berikut: Pertama, semua teks dari kolom tweet pada data\_df digabungkan menjadi satu string dengan metode join(), yang menggabungkan teks dalam bentuk list menjadi sebuah string panjang. Kemudian, objek WordCloud diciptakan dengan ukuran tertentu (lebar 800 dan tinggi 400) dan diisi dengan teks gabungan tadi melalui metode generate(). Akhirnya, menggunakan matplotlib, Word Cloud ditampilkan dengan imshow() untuk memvisualisasikannya, dan axis('off') digunakan untuk menghilangkan sumbu sehingga fokus hanya pada Word Cloud.

Word Cloud Visualization: Gambar pertama adalah hasil dari visualisasi Word Cloud yang menunjukkan kata-kata paling umum dari kolom tweet dalam ukuran dan warna yang berbeda, sesuai dengan frekuensi kemunculannya. Kata-kata seperti "USER," "dan," "yang," dan "Indonesia" muncul besar, menunjukkan bahwa kata-kata tersebut sering ditemukan di dalam tweet. Beberapa karakter seperti x9f, x80, x98 juga muncul, kemungkinan ini adalah encoding karakter yang tidak terkonversi dengan baik atau mewakili karakter khusus dalam teks asli.

1. Langkah 9: Meencari Korelasi antara data

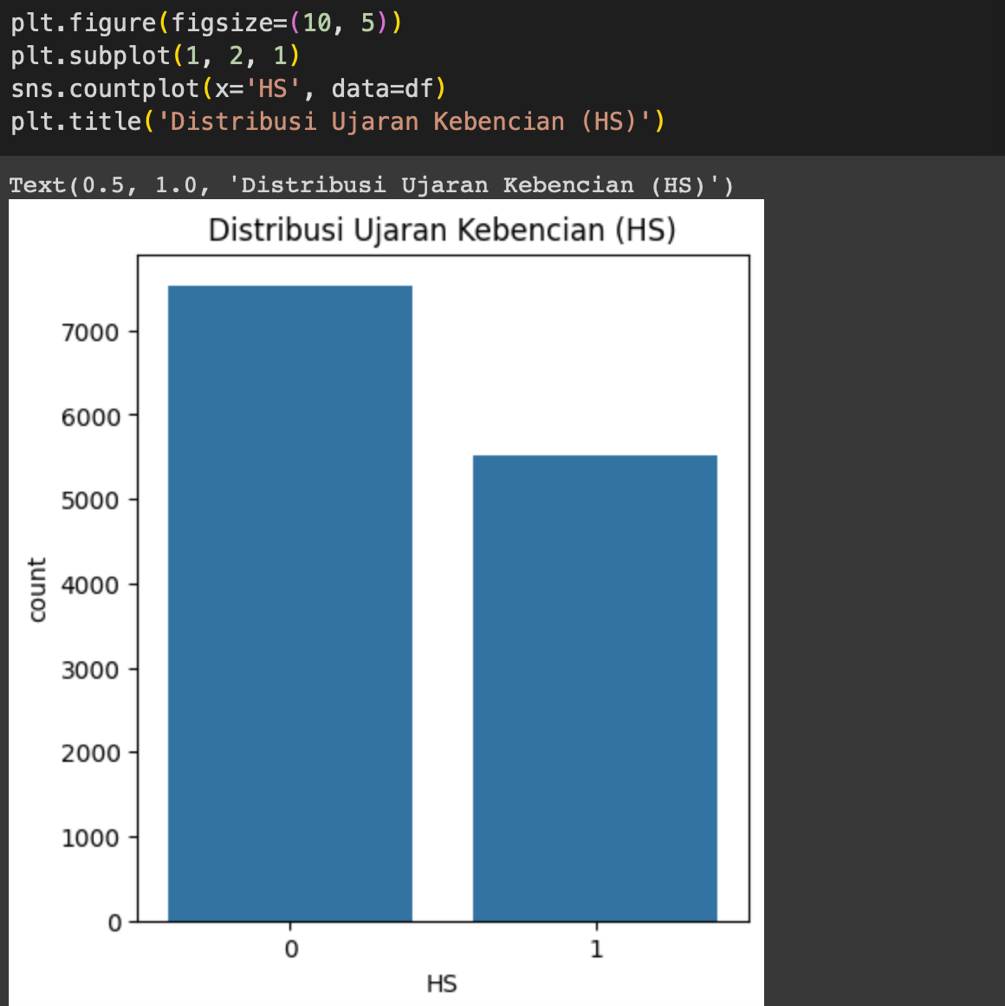




Kode Histogram:Kode ini secara otomatis membuat histogram untuk setiap kolom numerik dalam tiga DataFrame: new\_kamusalay\_df, data\_df, dan abusive\_df. Untuk setiap DataFrame, kode melakukan iterasi pada setiap kolom, dan jika kolom tersebut memiliki tipe data numerik (diperiksa dengan pd.api.types.is\_numeric\_dtype()), maka histogram akan dibuat untuk kolom tersebut menggunakan plt.hist(). Label sumbu-x diatur berdasarkan nama kolom, dan sumbu-y dilabeli "Frequency" untuk menunjukkan frekuensi nilai tersebut muncul. Setiap grafik juga diberi judul yang menunjukkan distribusi kolom dalam DataFrame terkait.

Contoh Histogram untuk Kolom 'HS' dalam data\_df: Grafik ini menunjukkan distribusi dari kolom HS dalam data\_df. Kolom HS adalah biner (0 atau 1), menunjukkan apakah suatu tweet mengandung ujaran kebencian. Pada sumbu-x, terdapat dua nilai, 0 dan 1. Nilai 0 menunjukkan tweet tanpa ujaran kebencian, sementara nilai 1 menunjukkan tweet dengan ujaran kebencian. Pada sumbu-y, frekuensi masing-masing nilai ditunjukkan, dengan jumlah tweet tanpa ujaran kebencian (0) lebih tinggi daripada yang mengandung ujaran kebencian (1). Ini menunjukkan bahwa sebagian besar tweet dalam dataset ini tidak mengandung ujaran kebencian.

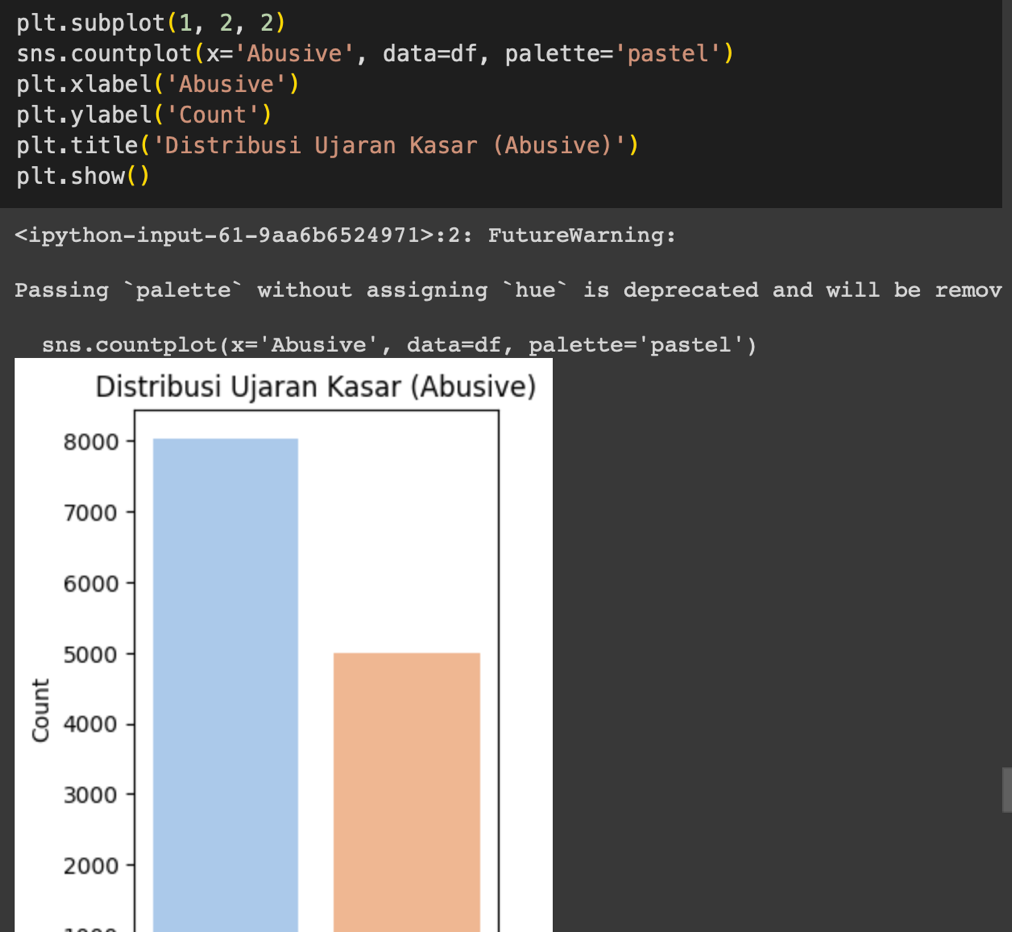
1. Langkah 10 : Korelasi Distribusi Ujaran Kebencian



Kode Visualisasi: plt.figure(figsize=(10, 5)): Menentukan ukuran keseluruhan grafik. plt.subplot(1, 2, 1): Mengatur tata letak subplot untuk grafik ini. Dalam konteks ini, hanya satu subplot yang ditampilkan. sns.countplot(x='HS', data=df): Menggunakan fungsi countplot dari seaborn untuk menampilkan frekuensi atau jumlah data untuk masing-masing kategori di kolom HS. Nilai HS digunakan sebagai sumbu-x, dan frekuensi atau jumlah kemunculannya sebagai sumbu-y. plt.title('Distribusi Ujaran Kebencian (HS)'): Menambahkan judul pada grafik untuk menjelaskan bahwa ini adalah distribusi data terkait ujaran kebencian.

Interpretasi Grafik: Grafik menunjukkan bahwa ada dua kategori di kolom HS: 0 menunjukkan tweet tanpa ujaran kebencian, yang memiliki frekuensi lebih tinggi (sekitar 7.000 tweet). 1 menunjukkan tweet dengan ujaran kebencian, dengan jumlah yang lebih rendah (sekitar 6.000 tweet). Distribusi ini menunjukkan bahwa sebagian besar data tidak mengandung ujaran kebencian, namun ada jumlah yang cukup signifikan dari data yang dikategorikan sebagai ujaran kebencian.

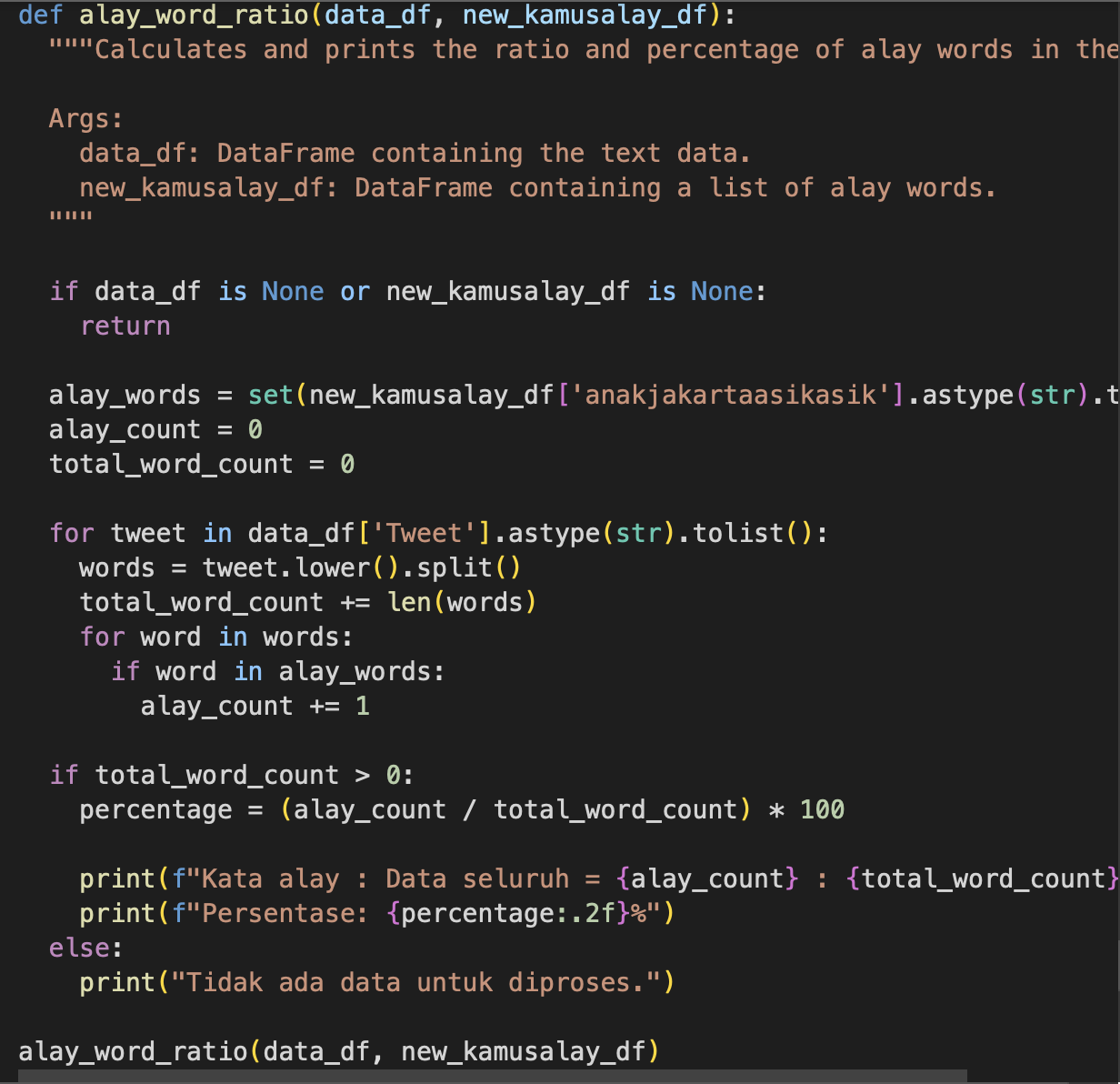
1. Langkah 11 : Distrubusi Ujaran Kasar

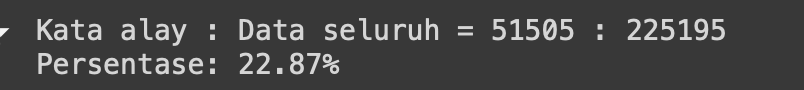


Kode Visualisasi: plt.subplot(1, 2, 2): Kode ini mengatur subplot untuk menempatkan grafik ini di kolom kedua dari dua grafik (tata letak 1 baris, 2 kolom). sns.countplot(x='Abusive', data=df, palette='pastel'): Menggunakan seaborn untuk membuat grafik countplot dari kolom Abusive. Sumbu-x mewakili kategori biner dari Abusive (0 untuk tidak kasar, 1 untuk kasar), sementara sumbu-y mewakili jumlah tweet untuk masing-masing kategori. Palet warna pastel digunakan untuk visualisasi warna yang lembut. plt.xlabel('Abusive') dan plt.ylabel('Count'): Label untuk sumbu-x dan sumbu-y, masing-masing untuk menampilkan informasi kategori Abusive dan jumlah frekuensi data. plt.title('Distribusi Ujaran Kasar (Abusive)'): Menambahkan judul pada grafik untuk menjelaskan bahwa ini adalah distribusi terkait kategori Abusive.

Interpretasi Grafik: Grafik menunjukkan dua kategori di kolom Abusive: 0: Mewakili tweet yang tidak mengandung ujaran kasar, dengan jumlah sekitar 8.000 tweet. 1: Mewakili tweet yang mengandung ujaran kasar, dengan jumlah sekitar 4.000 tweet. Data menunjukkan bahwa sebagian besar tweet dalam dataset tidak mengandung ujaran kasar, meskipun jumlah tweet dengan ujaran kasar juga cukup signifikan.

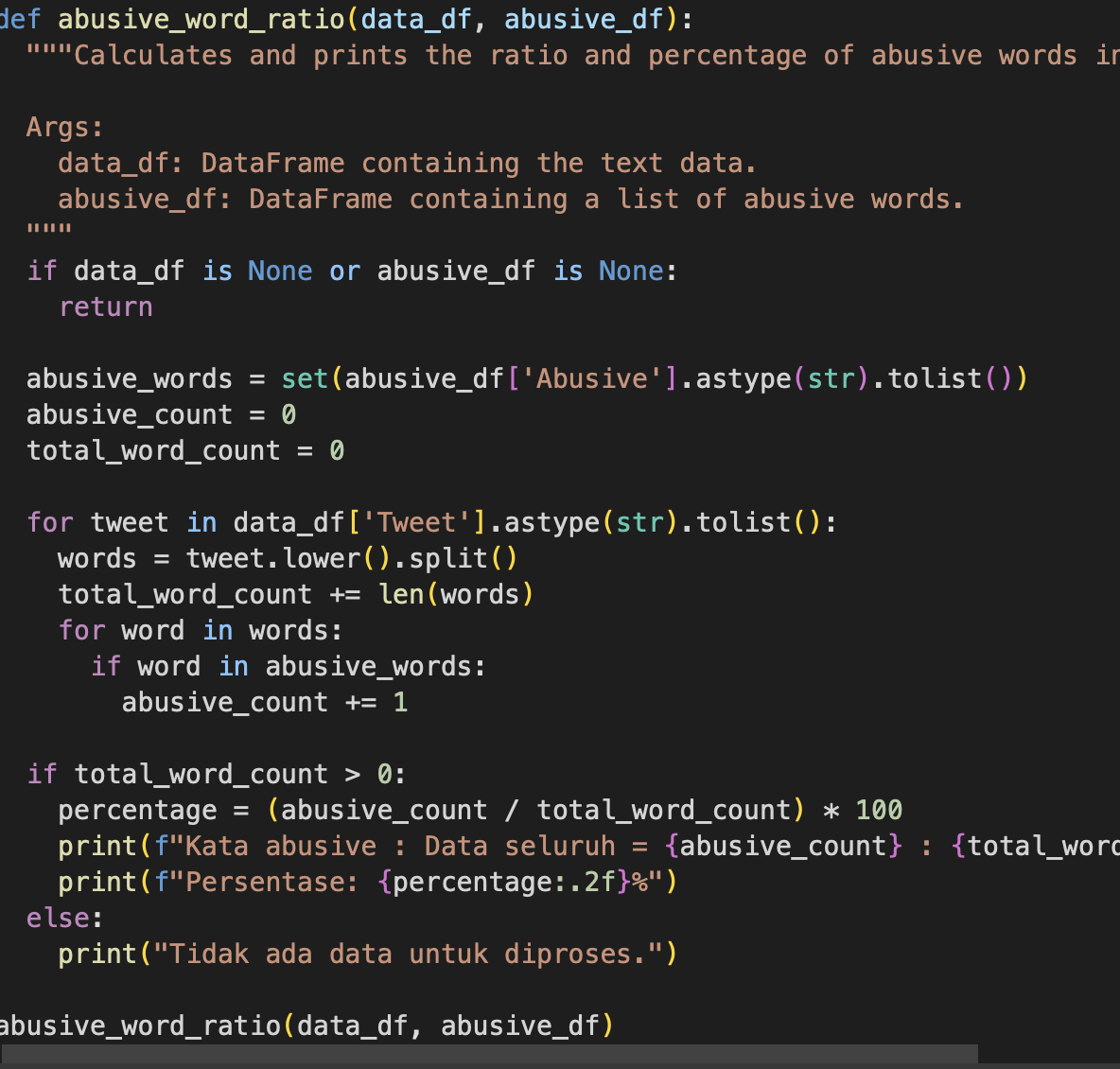
1. Langkah 12 : Perbandingan Kata alay dengan Data

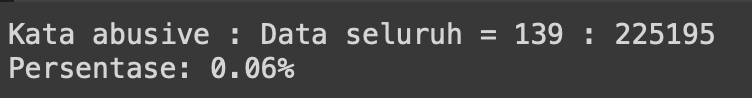




fungsi alay\_word\_ratio dalam kode ini digunakan untuk menghitung rasio dan persentase penggunaan kata alay dalam data tweet dengan menggunakan daftar kata alay dari DataFrame new\_kamusalay\_df. Fungsi ini terlebih dahulu memeriksa keberadaan DataFrame input; jika salah satu tidak tersedia, fungsi tidak akan dijalankan. Kemudian, kata alay dari kolom 'anakjakartaasikasik' di new\_kamusalay\_df diubah menjadi set, yang mempercepat proses pencarian. Dua variabel, yaitu alay\_count untuk menghitung jumlah kata alay dan total\_word\_count untuk jumlah total kata, diinisialisasi sebagai 0. Fungsi ini memproses setiap tweet dalam data\_df dengan mengubah teks menjadi huruf kecil, memisahkannya menjadi daftar kata, dan memperbarui total\_word\_count berdasarkan jumlah kata dalam tweet. Setiap kata dalam tweet diperiksa apakah ada di set kata alay; jika ada, alay\_count akan bertambah. Setelah semua tweet diproses, persentase kata alay dihitung dengan membagi alay\_count dengan total\_word\_count dan mengalikannya dengan 100. Hasil akhirnya menunjukkan bahwa terdapat 51.505 kata alay dari total 225.195 kata dalam dataset, dengan persentase 22,87%, yang mengindikasikan bahwa sekitar 22,87% dari kata-kata dalam dataset ini adalah kata alay.

1. Langkah 13 :





Fungsi abusive\_word\_ratio menghitung rasio dan persentase kata kasar dalam tweet menggunakan daftar kata kasar dari DataFrame abusive\_df. Pertama, kata-kata kasar dari kolom 'Abusive' diubah menjadi set untuk pencarian cepat, lalu dua variabel diinisialisasi untuk menghitung jumlah total kata dan jumlah kata kasar. Fungsi ini memproses setiap tweet dalam data\_df, memisahkannya menjadi kata-kata, lalu memeriksa setiap kata terhadap daftar kata kasar; jika ditemukan kecocokan, jumlah kata kasar bertambah. Setelah semua tweet dianalisis, persentase kata kasar dihitung dan hasilnya dicetak. Hasil menunjukkan bahwa dari total 225.195 kata dalam dataset, hanya 139 kata atau 0,06% yang merupakan kata kasar, menunjukkan frekuensi kata kasar yang rendah dalam dataset.

# **Hasil dan Kesimpulan**

**Hasil**

Berdasarkan analisis yang dilakukan, metode pembersihan data pada dataset tweet berhasil diterapkan untuk mengidentifikasi, mengonversi, dan menghapus kata-kata alay dan kasar. Hasilnya menunjukkan bahwa sekitar 22,87% dari kata-kata dalam dataset tergolong sebagai kata alay, yang kemudian dikonversi menjadi kata-kata formal menggunakan kamus alay. Proses ini membuat data lebih mudah dipahami dan dianalisis dalam konteks formal. Di sisi lain, analisis menunjukkan bahwa hanya 0,06% dari total kata yang terdeteksi sebagai kata kasar. Meskipun jumlahnya kecil, penghapusan kata kasar ini penting untuk menjaga validitas dan etika analisis data, mengurangi bias negatif dalam hasil akhir, dan memberikan gambaran yang lebih netral. Selain itu, metode yang digunakan memungkinkan penyimpanan data yang telah diproses ke dalam database SQLite, yang mempermudah pengelolaan, pemantauan, dan akses terhadap data hasil pembersihan untuk kebutuhan analisis lanjutan.

**Kesimpulan**

Penelitian ini berhasil menjawab rumusan masalah dengan menyediakan mekanisme pembersihan data yang efektif untuk menghilangkan elemen bahasa tidak formal dan kasar, menjadikan dataset lebih relevan dan sesuai untuk analisis berbasis Natural Language Processing (NLP) dan machine learning. Proses konversi kata alay menjadi kata formal meningkatkan kualitas data tanpa mengubah makna asli teks, sementara penghapusan kata kasar mendukung standar etika dan validitas data yang lebih baik. Penyimpanan hasil pembersihan dalam database SQLite juga mendukung efisiensi manajemen data, yang memungkinkan akses mudah untuk analisis lebih lanjut. Dengan demikian, metode ini memberikan kontribusi signifikan terhadap pengembangan analisis data teks yang berkualitas tinggi, membuka peluang untuk penerapan yang lebih luas dalam analitik sosial media, pemantauan konten, dan pengembangan model bahasa berbasis data.