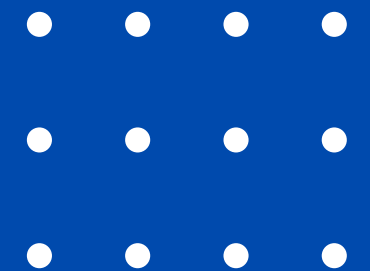


Data Text Media Analytics

Coldplay

27 Mei 2023 hingga 19 Juni 2023 dari website detik.com



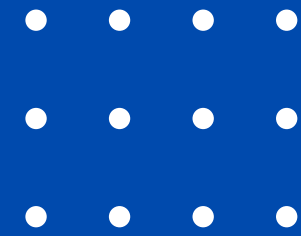
Anggota :

1. Ferdinand Sitompul (2206051576)
2. Najwa Putri Faradila (2206051355)
3. Kayla Zahira A (2206053890)

Kelas EVDA-B

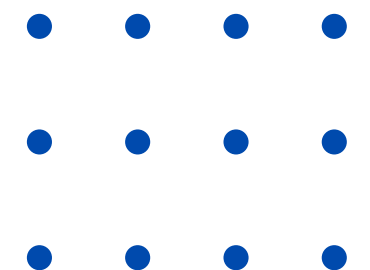


Pendahuluan



Kami melakukan pengambilan berita dari situs berita **detik.com** terkait pencarian dengan kata kunci “**coldplay**”. Kami mengumpulkan data dari 15 halaman berita, dimana pada setiap halaman pencarian akan dipindai untuk mendapatkan judul, tanggal, link, dan konten berita terkait kata kunci “**coldplay**”.

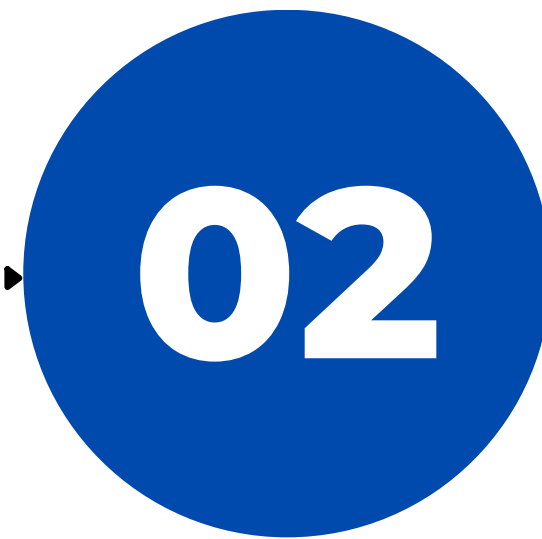
Data yang dikumpulkan nanti dapat kami gunakan untuk melakukan analisis terkait topik “coldplay,” seperti menganalisis tren berita terkait coldplay, pola pemberitaan, atau sentimen publik terhadap berita-berita tersebut.



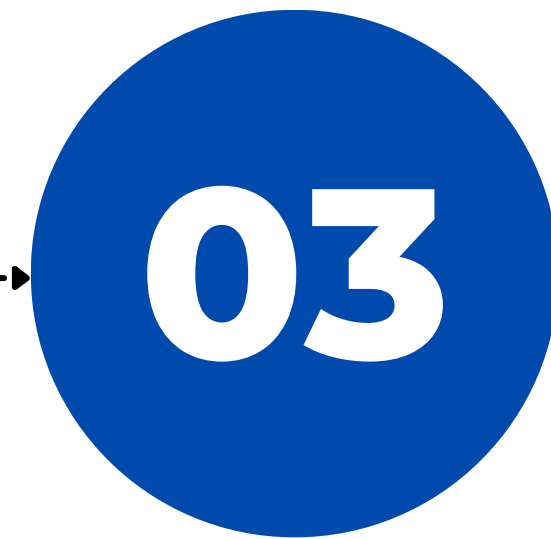
Daftar Isi



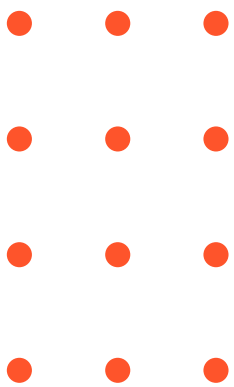
**Pengambilan
Data**



**Preprocessing
Data**



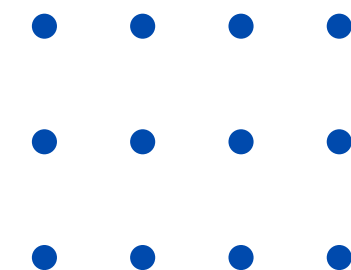
**Visualisasi
Data Text**





Mengimport Packages

```
import requests as req
from bs4 import BeautifulSoup as bs
from datetime import datetime
import csv
import pandas as pd
```



Proses Scrapping Data

Pengambilan data dilakukan dengan pemrograman sebagai berikut :

```
hades = {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36'}

# Pengambilan data dari situs web detik.com
def scrape_detik(hal):
    global hades
    a = 1
    for page in range(1, hal+1):
        url = f'https://www.detik.com/search/searchall?query=coldplay&siteid=2&page={page}'
        ge = req.get(url, headers=hades).text
        sop = bs(ge, 'lxml')
        li = sop.find('div', class_='list media_rows list-berita')
        lin = li.find_all('article')
        for x in lin:
            link = x.find('a')['href']
            date = x.find('a').find('span', class_='date').text.replace('WIB', '').replace('detikNews', '').split(',')[1]
            headline = x.find('a').find('h2').text
            ge_ = req.get(link, headers=hades).text
            sop_ = bs(ge_, 'lxml')
            content = sop_.find_all('div', class_='detail__body-text itp_bodycontent')
            for x in content:
                x = x.find_all('p')
                y = [y.text for y in x]
                content_ = ''.join(y).replace('\n', '').replace('ADVERTISEMENT', '').replace('SCROLL TO RESUME CONTENT', '')
                print(f'done[{a}] > {headline[0:10]}')
                a += 1
            with open('coldplay.csv', 'a') as file:
                wr = csv.writer(file, delimiter=',')
                wr.writerow([headline, date, link, content_])

scrape_detik(15)
```

1. Lakukan Pengambilan data
2. Setelah data diambil, simpan hasil scrapping kedalam bentuk csv
3. Simpan csv dan beri nama coldplay.csv

Preprocessing Data

1. Beri Nama Kolom untuk tiap kategori data

```
import pandas as pd

# Membaca data dari file CSV dengan menamai kolom
data = pd.read_csv('coldplay.csv', names=['Headline', 'Date', 'Link', 'Content'])
```

2. Hapus kolom 'Link' dan 'Date'

```
columns_to_drop = ['Link', 'Date']
data1 = data.drop(columns_to_drop, axis=1)

# Menampilkan dataframe setelah menghapus kolom
data1.head()
```

3. Periksa data teratas dan terbawah

```
# Menampilkan dataframe setelah menghapus kolom
data1.head()

data1.tail()
```

4. Pembersihan kata-kata yang kurang penting atau tidak unik

```
import csv
import re

def remove_common_words(text):
    common_words = ['ada', 'adalah', 'adanya', 'adapun', 'agak', 'agaknya', 'agar', 'akan', 'akankah', 'akhirnya', 'aku',
                    'akulah', 'amat', 'amatlah', 'anda', 'andalah', 'antar', 'antara', 'antaranya', 'apa', 'apaan', 'apabila',
                    'apakah', 'apalagi', 'apatah', 'atau', 'ataukah', 'ataupun', 'bagai', 'bagaikan', 'bagaimana',
                    'bagaimanakah', 'bagaimanapun', 'bagi', 'bahkan', 'bahwa', 'bahwasanya', 'banyak', 'beberapa', 'begini',
                    'beginian', 'beginikah', 'beginilah', 'begitu', 'begitukah', 'begitulah', 'begitupun', 'belum', 'belumlah',
                    'sesudah', 'sesudahnya', 'setelah', 'seterusnya', 'setiap', 'setidaknya', 'sewaktu', 'siapa', 'siapakah',
                    'siapapun', 'sini', 'sinilah', 'suatu', 'sudah', 'sudahkah', 'sudahlah', 'supaya', 'tadi', 'tadinya', 'tak',
                    'tanpa', 'tapi', 'telah', 'tentang', 'tentu', 'tentulah', 'tentunya', 'terdiri', 'terhadap', 'terhadapnya',
                    'terlalu', 'terlebih', 'tersebut', 'tersebutlah', 'tertentu', 'tetapi', 'tiap', 'tidak', 'tidakkah',
                    'tidaklah', 'toh', 'waduh', 'wah', 'wahai', 'walau', 'walaupun', 'wong', 'yaitu', 'yakni', 'yang', 'ya',
                    'www.detik.com']

    words = text.split()
    result = [word for word in words if word.lower() not in common_words]
    return ' '.join(result)
```


Visualisasi Data Teks

1. WordCloud

Program :

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Menggabungkan semua konten menjadi satu teks panjang
all_content = ' '.join(data1['Content'])

# Menghitung frekuensi kemunculan setiap kata
word_frequencies = {}
for word in all_content.split():
    if word not in word_frequencies:
        word_frequencies[word] = 1
    else:
        word_frequencies[word] += 1

# Mengurutkan kata-kata berdasarkan frekuensinya
sorted_words = sorted(word_frequencies.items(), key=lambda x: x[1], reverse=True)

# Memilih kata-kata dengan frekuensi tertinggi (misalnya, 100 kata teratas)
top_words = dict(sorted_words[:100])

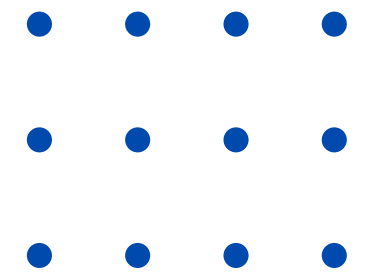
# Membuat objek WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate_from_frequencies

# Menampilkan Word Cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

Hasil Visualisasi Data :



Visualisasi Data Teks



2. Histogram

Program :

```
# Menggabungkan semua konten menjadi satu teks panjang
all_content = ' '.join(data1['Content'])

# Menghitung frekuensi kemunculan setiap kata
word_frequencies = {}
for word in all_content.split():
    if word not in word_frequencies:
        word_frequencies[word] = 1
    else:
        word_frequencies[word] += 1

# Mengurutkan kata-kata berdasarkan frekuensinya
sorted_words = sorted(word_frequencies.items(), key = lambda x: x[1], reverse = True)

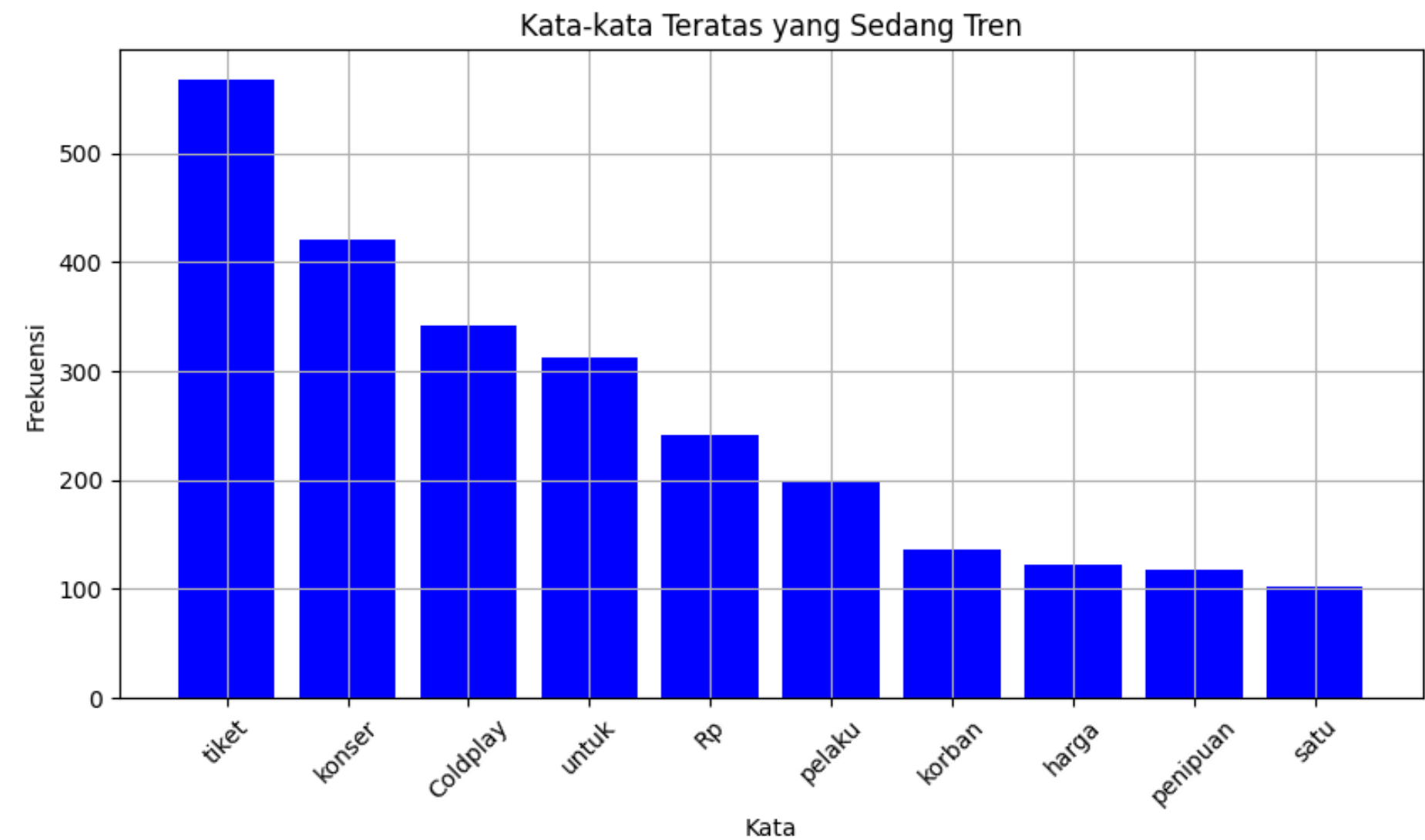
# Memilih kata-kata dengan frekuensi tertinggi (misalnya, 10 kata teratas)
top_words = dict(sorted_words[:10])

# Membuat DataFrame dari kata-kata dan frekuensinya
df = pd.DataFrame(list(top_words.items()), columns = ['Kata', 'Frekuensi'])

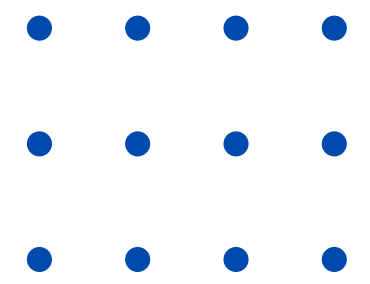
# Mengurutkan DataFrame berdasarkan frekuensi secara menurun
df = df.sort_values('Frekuensi', ascending = False)

# Membuat plot diagram histogram
plt.figure(figsize=(10, 5))
plt.bar(df['Kata'], df['Frekuensi'], color='blue')
plt.xlabel('Kata')
plt.ylabel('Frekuensi')
plt.title('Kata-kata Teratas yang Sedang Tren')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

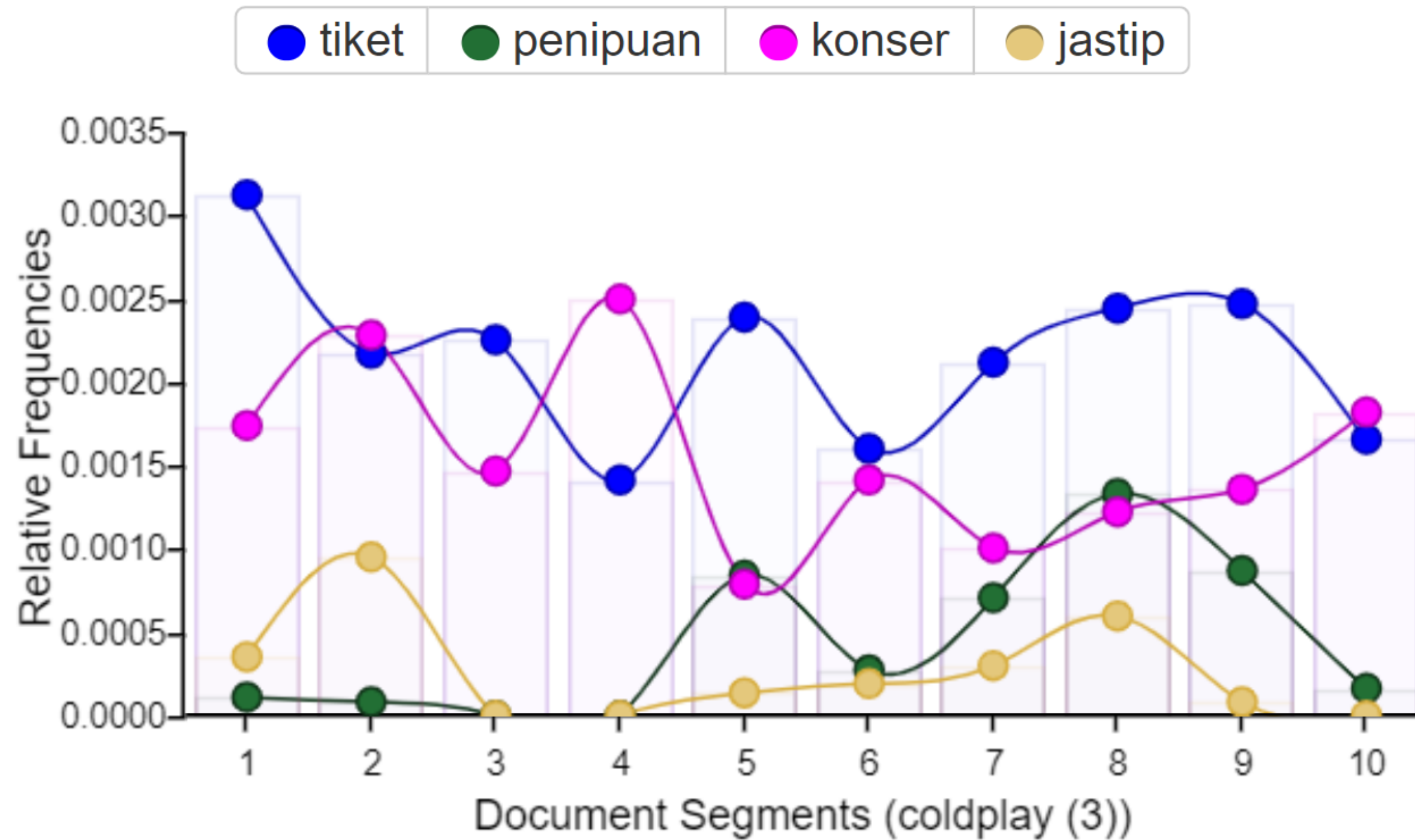
Hasil Visualisasi Data :



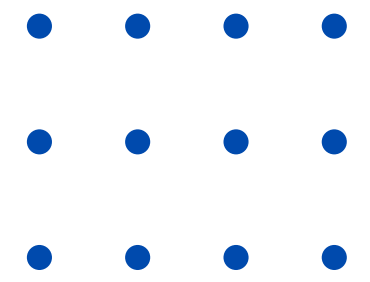
Visualisasi Data Teks



3. Trends



Visualisasi Data Teks



4. Sentimen Publik

Program :

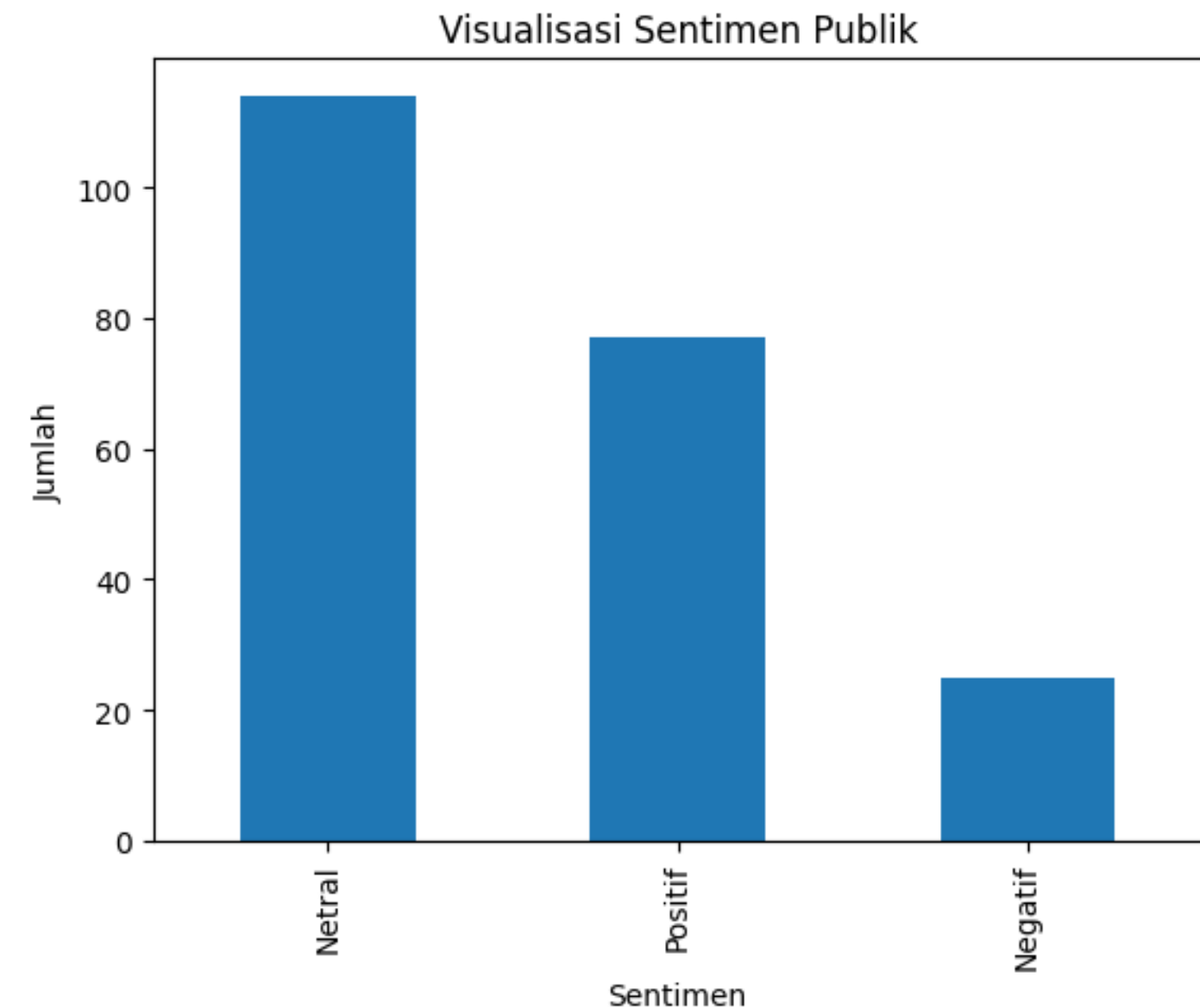
```
from textblob import TextBlob

# Mendapatkan sentimen teks
def get_sentiment(text):
    blob = TextBlob(text)
    sentiment = blob.sentiment.polarity
    if sentiment > 0:
        return 'Positif'
    elif sentiment < 0:
        return 'Negatif'
    else:
        return 'Netral'

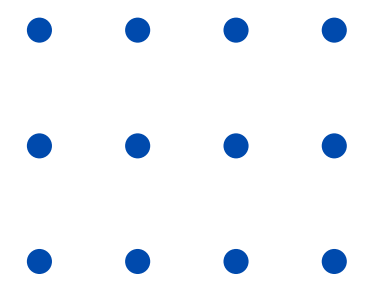
# Menerapkan analisis sentimen pada kolom 'Content'
data1['Sentiment'] = data1['Content'].apply(get_sentiment)

# Menampilkan visualisasi sentimen publik
sentiment_counts = data1['Sentiment'].value_counts()
sentiment_counts.plot(kind = 'bar', xlabel = 'Sentimen',
                      ylabel = 'Jumlah', title = 'Visualisasi Sentimen Publik')
plt.show()
```

Hasil Visualisasi Data :



Kesimpulan



Kedatangan Coldplay ke Indonesia menjadi topik yang hangat diperbincangkan di banyak media, termasuk di portal berita detik.com. Setelah dilakukan analisis data, terlihat beberapa kata kunci yang sering muncul yang berkaitan dengan topik ini seperti "**tiket**", "**Coldplay**", dan "**konser**".

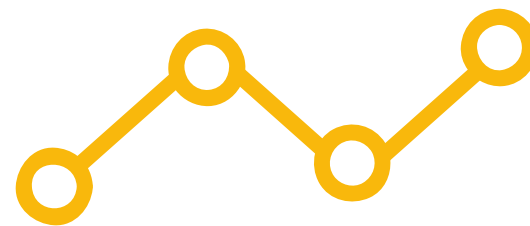
Banyak masyarakat yang antusias menyambut konser Coldplay ini, sebagaimana tercermin dalam grafik sentimen publik yang menunjukkan adanya **sentimen positif** yang cukup tinggi, berada **di urutan kedua**.

Namun, di sisi lain, terdapat juga **sentimen negatif** yang muncul. Hal ini dapat dilihat dari grafik trend yang menunjukkan hubungan antara kata-kata "**penipuan**" dan "**jastip**". Pola grafik yang berbanding lurus tersebut menunjukkan adanya perbincangan mengenai kasus penipuan dalam penjualan tiket konser Coldplay, yang disebabkan oleh adanya jasa titip ("jastip") yang tidak dapat dipercaya dan diandalkan.



Meskipun terdapat sentimen positif dan negatif terhadap kedatangan Coldplay dan konser mereka, **mayoritas** masyarakat pun pada akhirnya cenderung memiliki **sentimen netral**.





TERIMA KASIH

