

BAB V

KESIMPULAN

5.1 Kesimpulan Utama

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya mengenai perilaku informasi pengguna Museum Nasional Indonesia dalam ulasan Google Maps melalui analisis sentimen, dapat diambil kesimpulan sebagai berikut:

1. Distribusi sentimen ulasan pada periode sebelum kebakaran menunjukkan dominasi sentimen positif sebesar 80,61% (1023 ulasan), sedangkan sentimen negatif sebesar 19,39% (246 ulasan). Temuan ini menunjukkan bahwa sebelum kebakaran, evaluasi pengunjung terhadap pengalaman kunjungan cenderung positif dan umumnya menilai layanan museum memadai.
2. Distribusi sentimen pasca revitalisasi masih didominasi sentimen positif sebesar 72,38% (1140 ulasan), namun sentimen negatif meningkat menjadi 27,62% (435 ulasan). Jika dibandingkan antarperiode, terjadi penurunan proporsi sentimen positif sebesar 8,23 poin persentase dan peningkatan sentimen negatif sebesar 8,23 poin persentase. Berdasarkan teori *expectancy-disconfirmation* pergeseran ini menunjukkan bahwa setelah revitalisasi, terdapat ketidaksesuaian harapan dan pengalaman pada sebagian pengunjung sehingga memunculkan penilaian yang lebih kritis.
3. Ditinjau dari aspek layanan dominan, terjadi perubahan fokus evaluasi pengunjung antarperiode. Sebelum kebakaran, aspek yang paling dominan adalah koleksi dengan 466 data, disusul fasilitas 369 data, aksesibilitas 256 data, dan pelayanan 178 data. Pasca revitalisasi, aspek dominan bergeser menjadi fasilitas dengan 504 data, disusul koleksi 467 data, aksesibilitas 328 data, dan pelayanan 276 data.
4. Hasil klasifikasi sentimen menggunakan Support Vector Machine menunjukkan kinerja yang memadai, dengan akurasi 85% pada periode sebelum kebakaran dan 80% pada periode pasca revitalisasi pada pembagian data 80% latih dan 20% uji. Confusion matrix menunjukkan model lebih kuat mengenali ulasan positif dibanding ulasan negatif,

sementara kesalahan klasifikasi lebih sering terjadi pada ulasan yang ekspresinya ambigu atau memuat penilaian campuran.

Berdasarkan temuan tersebut, dapat disimpulkan bahwa ulasan Google Maps mampu merekam evaluasi pengguna sekaligus menggambarkan *prises* pemakaian pengalaman kunjungan. Dominasi sentimen positif menunjukkan bahwa Museum Nasional Indonesia tetap dipersepsikan baik secara umum, namun peningkatan sentimen negatif dan pergeseran aspek dominan ke fasilitas menandakan meningkatnya ekspektasi serta munculnya penilaian yang lebih kritis terhadap aspek layanan penunjang pengalaman informasi. Dalam perspektif *sense-making*, perubahan aspek dominan mencerminkan pergeseran *gap* dan *bridge* yang dianggap penting oleh pengunjung, sedangkan dalam perspektif *expectancy-disconfirmation*, perubahan proporsi sentimen mengindikasikan ketidaksesuaian harapan dan pengalaman pada sebagian pengunjung

5.2 Keterbatasan Penelitian

Peneliti menyadari bahwa penelitian ini masih memiliki beberapa keterbatasan yang dapat memengaruhi keluasan penafsiran hasil. Adapun keterbatasan tersebut antara lain:

1. Penelitian ini hanya menggunakan sumber data berupa ulasan pada platform Google Maps untuk objek Museum Nasional Indonesia. Temuan penelitian merepresentasikan persepsi pengguna yang memilih menuliskan ulasan di Google Maps, sehingga tidak dapat sepenuhnya mewakili pengalaman pengunjung yang tidak menulis ulasan, maupun yang menyampaikan penilaian melalui platform lain atau secara luring.
2. Data yang dianalisis terbatas pada ulasan berbentuk teks sesuai kriteria periode penelitian. Ulasan yang hanya berupa rating tanpa teks tidak dianalisis, padahal rating dapat merefleksikan penilaian pengguna. Selain itu, ulasan teks yang sangat singkat atau minim konteks dapat membatasi ketepatan interpretasi terhadap alasan kepuasan atau ketidakpuasan pengguna.
3. Penelitian ini memetakan kecenderungan sentimen melalui analisis sentimen dengan pendekatan *lexicon-based* dan klasifikasi SVM. Pendekatan tersebut

masih memiliki keterbatasan dalam menangkap ekspresi bahasa yang ambigu, ironi, sarkasme, atau penilaian campuran, sehingga terdapat kemungkinan sebagian ulasan tidak terklasifikasi sesuai maksud penulis ulasan.

4. Fokus penelitian berada pada distribusi sentimen dan aspek layanan dominan yang muncul dalam ulasan. Faktor lain yang dapat memengaruhi persepsi dan kepuasan pengunjung, seperti motivasi kunjungan, latar belakang pengunjung, kepadatan pengunjung pada hari tertentu, kebijakan pengelola, atau kondisi situasional saat kunjungan, tidak dapat diidentifikasi secara lengkap karena tidak selalu muncul dalam teks ulasan.
5. Pengumpulan data dilakukan dalam rentang waktu tertentu, sehingga hasil penelitian menggambarkan kecenderungan persepsi pengguna pada periode tersebut. Persepsi publik dapat berubah seiring perubahan kebijakan museum, pembaruan fasilitas, strategi komunikasi digital, maupun dinamika komentar pengguna di kemudian hari.

5.3 Saran Untuk Penelitian Selanjutnya

Berdasarkan hasil penelitian dan keterbatasan yang telah diuraikan, beberapa saran yang dapat dipertimbangkan untuk penelitian selanjutnya adalah sebagai berikut:

1. Penelitian selanjutnya dapat memperluas sumber data dengan menggabungkan ulasan dari platform lain, seperti Google Review pada hasil pencarian, TripAdvisor, Instagram, atau X, agar gambaran persepsi pengguna tidak hanya bergantung pada satu platform dan dapat dibandingkan antar kanal digital.
2. Penelitian berikutnya disarankan menganalisis rating bintang bersamaan dengan teks ulasan. Penggabungan keduanya dapat membantu melihat konsistensi antara penilaian numerik dan narasi, serta mengurangi bias dari ulasan yang sangat singkat atau tidak eksplisit dalam menyatakan sentimen.
3. Untuk meningkatkan ketepatan klasifikasi sentimen, penelitian selanjutnya dapat menguji model berbasis deep learning atau transformer berbahasa Indonesia, serta membandingkannya dengan SVM dan lexicon-based. Dengan cara ini, dapat diketahui metode yang paling sesuai untuk menangkap

variasi bahasa ulasan, termasuk ungkapan halus, ambiguitas, atau penilaian campuran



DAFTAR PUSTAKA

- Ahadi, A., Singh, A., Bower, M., & Garrett, M. (2022). Text mining in education: A bibliometrics-based systematic review. *Education Sciences*, 12(3), 1–17. <https://doi.org/10.3390/educsci12030210>
- ASTC. (2022). *Museum digital engagement: Before, during, and beyond the COVID-19 pandemic*. https://www.astc.org/wp-content/uploads/2022/05/ASTC_Museum-Digital-Engagement_Before-During-and-Beyond-the-COVID-19-Pandemic.pdf
- Bahri, C. A., & Suadaa, L. H. (2023). Aspect-based sentiment analysis in Bromo Tengger Semeru National Park Indonesia based on Google Maps user reviews. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 17(1), 79–90. <https://doi.org/10.22146/ijccs.77354>
- Chen, C., & Chang, C.-C. (2024). *Evaluating public library services through user-generated content: Social network analysis of Google Maps reviews*. <https://doi.org/10.20944/preprints202405.0619.v1>
- Dervin, B. (1992). From the mind's eye of the user: The sense-making qualitative-quantitative methodology. In *Qualitative research in information management*. Libraries Unlimited.
- Handayani, P. K. (2021). Penerapan algoritma Support Vector Machine (SVM) untuk analisis pola klasifikasi pada parkinson's dataset. *Indonesian Journal of Technology, Informatics and Science (IJTIS)*, 3(1), 31–35. <https://doi.org/10.24176/ijtis.v3i1.7530>
- Hernon, P., Altman, E., & Dugan, R. E. (2015). *Assesing service quality: Satisfying the expectations of library customers* (Third Edition). American Library Association.
- International Council of Museums. (2022). ICOM approves a new museum definition. *ICOM*.

- International Council of Museums. (2025). Standards and guidelines. *ICOM DOCUMENTATION*. <https://cidoc.mini.icom.museum/standards/cidoc-standards-guidelines/>
- Ipmawati, J., Saifulloh, S., & Kusnawi, K. (2024). Analisis sentimen tempat wisata berdasarkan ulasan pada Google Maps menggunakan algoritma Support Vector Machine: Sentiment analysis of tourist attractions based on reviews on Google Maps Using the Support Vector Machine Algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 247–256. <https://doi.org/10.57152/malcom.v4i1.1066>
- Ismagilova, E., Slade, E. L., Rana, N. P., & Dwivedi, Y. K. (2020). The effect of electronic word of mouth communications on intention to buy: A meta-analysis. *Information Systems Frontiers*, 22(5), 1203–1226. <https://doi.org/10.1007/s10796-019-09924-y>
- Jha, A. K., & Shah, S. (2021). Disconfirmation effect on online review credibility: An experimental analysis. *Decision Support Systems*, 145, 1–12. <https://doi.org/10.1016/j.dss.2021.113519>
- Jiang, G., Liu, F., Liu, W., Liu, S., Chen, Y., & Xu, D. (2021). Effects of information quality on information adoption on social media review platforms: Moderating role of perceived risk. *Data Science and Management*, 1(1), 13–22. <https://doi.org/10.1016/j.dsm.2021.02.004>
- Kwiecien, K., Chansanam, W., & Tuamsuk, K. (2025). Digital cultural heritage in Southeast Asia: Knowledge structures and resources in GLAM Institutions. *Informatics*, 12(3), 1–22. <https://doi.org/10.3390/informatics12030096>
- Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions* (2nd edn). Cambridge University Press. https://books.google.co.id/books?id=PdX7DwAAQBAJ&printsec=copyright&hl=id&source=gbs_pub_info_r#v=onepage&q&f=false
- Miles, M. B., Huberman, A. M., & Saldana, J. (2020). *Qualitative data analysis: A methods sourcebook* (4th Edition). SAGE.

- Montejo-Ráez, A., & Jiménez-Zafra, S. M. (2022). Current approaches and applications in Natural Language Processing. *Applied Sciences*, 12(10), 1–6. <https://doi.org/10.3390/app12104859>
- Oh, D.-G. (2020). Beyond providing information: An analysis on the perceived service quality, satisfaction, and loyalty of public library customers. *Libri*, 70(4), 345–359. <https://doi.org/10.1515/libri-2020-0006>
- Oliver, R. L. (1980). A Cognitive Model of the Antecedents and Consequences of Satisfaction Decisions. *Journal of Marketing Research*, 17(4), 460–469. <https://doi.org/10.2307/3150499>
- OpenStax. (2020). *Psychology 2e* (2nd edn). Rice University. <https://openstax.org/details/books/psychology-2>
- Paul, H., & Nikolaev, A. (2021). Fake review detection on online E-commerce platforms: A systematic literature review. *Data Mining and Knowledge Discovery*, 35(5), 1830–1881. <https://doi.org/10.1007/s10618-021-00772-6>
- Rahmi, L., Fakhlina, R. J., & Daulay, R. (2023). Penggunaan media sosial dan perilaku pencarian informasi mahasiswa Ilmu Perpustakaan dan Informasi dalam pembelajaran. *Literatify : Trends in Library Developments*, 286–298. <https://doi.org/10.24252/literatify.vi.42721>
- Saputra, I., & Kristiyanti, D. A. (2022). *Machine Learning untuk Pemula*. Informatika.
- Schiebler, T., Lee, N., & Brodbeck, F. C. (2025). Expectancy-disconfirmation and consumer satisfaction: A meta-analysis. *Journal of the Academy of Marketing Science*. <https://doi.org/10.1007/s11747-024-01078-x>
- Semary, N. A., Ahmed, W., Amin, K., Pławiak, P., & Hammad, M. (2024). Enhancing machine learning-based sentiment analysis through feature extraction techniques. *PLOS ONE*, 19(2), 1–19. <https://doi.org/10.1371/journal.pone.0294968>
- Sugiyono. (2023). *Metode penelitian kuantitatif kualitatif dan R&D* (Kedua). Alfabeta.

- Tan, K. L., Lee, C. P., & Lim, K. M. (2023). A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. *Applied Sciences*, 13(7), 1–21. <https://doi.org/10.3390/app13074550>
- Teoh, T. T., & Rong, Z. (with Teoh, T. T., & Rong, Z.). (2022). *Text Mining*. Springer Singapore. https://doi.org/10.1007/978-981-16-8615-3_14
- Urquhart, C., Cheuk, B., Lam, L., & Snowden, D. (2023). Sense-making, sensemaking and sense making—A systematic review and meta-synthesis of literature in information science and education: An Annual Review of Information Science and Technology (ARIST) paper. *Journal of the Association for Information Science and Technology*, 76(1), 3–97. <https://doi.org/10.1002/asi.24866>
- Verma, D., Dewani, P. P., Behl, A., & Dwivedi, Y. K. (2023). Understanding the impact of eWOM communication through the lens of information adoption model: A meta-analytic structural equation modeling perspective. *Computers in Human Behavior*, 143, 1–16. <https://doi.org/10.1016/j.chb.2023.107710>
- Yadav, N., Verma, S., & Chikhalkar, R. (2024). Online reviews towards reducing risk. *Journal of Tourism Futures*, 10(2), 299–316. <https://doi.org/10.1108/JTF-01-2022-0016>
- Yu, L., & Liu, Y. (2022). Information experience as an object of LIS research: A definition based on concept analysis. *Journal of Documentation*, 78(6), 1487–1508. <https://doi.org/10.1108/JD-10-2021-0195>
- Zaffran, I. R., & Santi, R. C. N. (2025). Klasifikasi sentimen wisatawan Museum Gedung Juang 45 Bekasi berdasarkan ulasan Google Maps dengan algoritma KNN. *Jurnal Mahasiswa Teknik Informatika*, 9(5), 8370–8377. <https://doi.org/https://doi.org/10.36040/jati.v9i5.15111>



TEXT PREPROCESSING

PROSES CLEANING

```
import pandas as pd
```

```
df = pd.read_csv("Periode Sebelum.csv")
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1334 entries, 0 to 1333
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0    text    1334 non-null    object
dtypes: object(1)
memory usage: 10.6+ KB
```

	text
0	Akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.
2	Sangat bagus karena banyak edukasi untuk anak2...
3	Koleksinya lengkap dari seluruh Provinsi di In...
4	Pas akhir pekan jadi ramai dengan pengunjung y...

```
df = pd.DataFrame(df[['text']])
df.info()
df.head(5)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1334 entries, 0 to 1333
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0    text    1334 non-null    object
dtypes: object(1)
memory usage: 10.6+ KB
```

	text
0	Akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.
2	Sangat bagus karena banyak edukasi untuk anak2...
3	Koleksinya lengkap dari seluruh Provinsi di In...
4	Pas akhir pekan jadi ramai dengan pengunjung y...

```
#menghapus data duplikat
```

```
df.drop_duplicates(subset ="text", inplace = True)
```

```
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1280 entries, 0 to 1333
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0    text    1280 non-null    object
dtypes: object(1)
memory usage: 20.0+ KB
```

	text
0	Akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.
2	Sangat bagus karena banyak edukasi untuk anak2...
3	Koleksinya lengkap dari seluruh Provinsi di In...
4	Pas akhir pekan jadi ramai dengan pengunjung y...

```

import re
import string
import nltk

# Fungsi untuk menghapus URL
def remove_URL(tweet):
    if tweet is not None and isinstance(tweet, str):
        url = re.compile(r'https?://\S+|www\.\S+')
        return url.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus HTML
def remove_html(tweet):
    if tweet is not None and isinstance(tweet, str):
        html = re.compile(r'<.*?>')
        return html.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus emoji
def remove_emoji(tweet):
    if tweet is not None and isinstance(tweet, str):
        emoji_pattern = re.compile("["
            u"\U0001F600-\U0001F64F" # emoticons
            u"\U0001F300-\U0001F5FF" # symbols & pictographs
            u"\U0001F680-\U0001F6FF" # transport & map symbols
            u"\U0001F700-\U0001F77F" # alchemical symbols
            u"\U0001F780-\U0001F7FF" # Geometric Shapes Extended
            u"\U0001F800-\U0001F8FF" # Supplemental Arrows-C
            u"\U0001F900-\U0001F9FF" # Supplemental Symbols and Pictographs
            u"\U0001FA00-\U0001FA6F" # Chess Symbols
            u"\U0001FA70-\U0001FAFF" # Symbols and Pictographs Extended-A
            u"\U0001F004-\U0001F0CF" # Additional emoticons
            u"\U0001F1E0-\U0001F1FF" # flags
            "]+", flags=re.UNICODE)
        return emoji_pattern.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus simbol
def remove_symbols(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'^a-zA-Z0-9\s', '', tweet)
        return tweet

# Fungsi untuk menghapus angka
def remove_numbers(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'\d', '', tweet)
        return tweet

# Fungsi hapus username
def remove_usernames(text):
    if text is not None and isinstance(text, str):
        return re.sub(r'@\w+', '', text)
    else:
        return text

df['cleaning'] = df['text'].apply(lambda x: remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_usernames(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))

df.head(5)

```

	text	cleaning
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...
3	Koleksinya lengkap dari seluruh Provinsi di In...	Koleksinya lengkap dari seluruh Provinsi di In...
4	Pas akhir pekan jadi ramai dengan pengunjung y...	Pas akhir pekan jadi ramai dengan pengunjung y...

PROSES CASE FOLDING

```
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(5)
```

	text	cleaning	case_folding
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...
3	Koleksinya lengkap dari seluruh Provinsi di In	Koleksinya lengkap dari seluruh Provinsi di	koleksinya lengkap dari seluruh provinsi di in

Normalisasi Kata

```
import pandas as pd
import requests
from io import BytesIO

# Fungsi penggantian kata tidak baku
def replace_taboo_words(text, kamus_tidak_baku):
    if isinstance(text, str):
        words = text.split()
        replaced_words = []
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []

        for word in words:
            if word in kamus_tidak_baku:
                baku_word = kamus_tidak_baku[word]
                if isinstance(baku_word, str) and all(char.isalpha() for char in baku_word):
                    replaced_words.append(baku_word)
                    kalimat_baku.append(baku_word)
                    kata_diganti.append(word)
                    kata_tidak_baku_hash.append(hash(word))
            else:
                replaced_words.append(word)
        replaced_text = ' '.join(replaced_words)
    else:
        replaced_text = ''
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []

    return replaced_text, kalimat_baku, kata_diganti, kata_tidak_baku_hash

# Baca dataset kamu (pastikan df sudah tersedia)
data = pd.DataFrame(df[['text', 'cleaning', 'case_folding']])
data.head()
```

	text	cleaning	case_folding
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...
3	Koleksinya lengkap dari seluruh Provinsi di In	Koleksinya lengkap dari seluruh Provinsi di	koleksinya lengkap dari seluruh provinsi di in

```
# Unduh dan baca kamus dari GitHub
url = "https://github.com/analysisdatasentiment/kamus_kata_baku/raw/main/kamuskatabaku.xlsx"
response = requests.get(url)
file_excel = BytesIO(response.content)
kamus_data = pd.read_excel(file_excel)

# Buat dictionary dari kamus
kamus_tidak_baku_dict = dict(zip(kamus_data['tidak_baku'], kamus_data['kata_baku']))
```

```
# Terapkan fungsi normalisasi
data[['normalisasi', 'Kata_Baku', 'Kata_Tidak_Baku', 'Kata_Tidak_Baku_Hash']] = data['case_folding'].apply(
    lambda x: pd.Series(replace_taboo_words(x, kamus_tidak_baku_dict))
)

# Ambil kolom yang relevan
df = pd.DataFrame(data[['text', 'cleaning', 'case_folding', 'normalisasi']])
df.head(20)
```

	text	cleaning	case_folding	normalisasi
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...
3	Koleksinya lengkap dari seluruh Provinsi di In...	Koleksinya lengkap dari seluruh Provinsi di In...	koleksinya lengkap dari seluruh provinsi di in...	koleksinya lengkap dari seluruh provinsi di in...
4	Pas akhir pekan jadi ramai dengan pengunjung y...	Pas akhir pekan jadi ramai dengan pengunjung y...	pas akhir pekan jadi ramai dengan pengunjung y...	pas akhir pekan jadi ramai dengan pengunjung y...
5	Kalau kesini, anda harus coba memakai guide ya...	Kalau kesini anda harus coba memakai guide yan...	kalau kesini anda harus coba memakai guide yan...	kalau kesini anda harus coba memakai guide yan...
6	Sangat bagus, untuk wisata edukasi.	Sangat bagus untuk wisata edukasi	sangat bagus untuk wisata edukasi	sangat bagus untuk wisata edukasi
7	Sangat menarik. Koleksi yang luar biasa yang m...	Sangat menarik Koleksi yang luar biasa yang me...	sangat menarik koleksi yang luar biasa yang me...	sangat menarik koleksi yang luar biasa yang me...
8	Menarik	Menarik	menarik	menarik
9	Sdh bbrp kl kami mengunjungi museum ini. Dan k...	Sdh bbrp kl kami mengunjungi museum ini Dan ka...	sdh bbrp kl kami mengunjungi museum ini dan ka...	sudah beberapa kalau kami mengunjungi museum i...
10	Sangat menyenangkan berkunjung ke museum nasio...	Sangat menyenangkan berkunjung ke museum nasio...	sangat menyenangkan berkunjung ke museum nasio...	sangat menyenangkan berkunjung ke museum nasio...
11	Kemarin kesini pas lg ada pameran korea, trs k...	Kemarin kesini pas lg ada pameran korea trs ke...	kemarin kesini pas lg ada pameran korea trs ke...	kemarin kesini pas lagi ada pameran korea teru...
12	Kesini waktu lagi ada pemeran keramik. Tempatn...	Kesini waktu lagi ada pemeran keramik Tempatny...	kesini waktu lagi ada pemeran keramik tempatny...	kesini waktu lagi ada pemeran keramik tempatny...
13	Museum dengan pola sirkulasi ramah anak karena...	Museum dengan pola sirkulasi ramah anak karena...	museum dengan pola sirkulasi ramah anak karena...	museum dengan pola sirkulasi ramah anak karena...
14	Tempat jogging favorit, sering ada event menarik	Tempat jogging favorit sering ada event menarik	tempat jogging favorit sering ada event menarik	tempat jogging favorit sering ada event menarik
15	keterlalu kalau nggak pernah kesini, akses t...	keterlalu kalau nggak pernah kesini akses tr...	keterlalu kalau nggak pernah kesini akses tr...	keterlalu kalau tidak pernah kesini akses tr...
16	Museum di gedung yang jadi satu dendaan arca-ar...	Museum di gedung yang jadi satu dendaan arcaarc...	museum di gedung yang jadi satu dendaan arcaarc...	museum di gedung yang jadi satu dendaan arcaarc...

TOKENIZATION

```
def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['normalisasi'].apply(tokenize)

df.head(5)
```

	text	cleaning	case_folding	normalisasi	tokenize
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	[akhirnya, bisa, ikut, berpartisipasi, untuk, ...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	[panas, gersang, sejak, pepohonan, ditebangin]
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	[sangat, bagus, karena, banyak, edukasi, untuk...
3	Koleksinya lengkap dari seluruh Provinsi di In...	Koleksinya lengkap dari seluruh Provinsi di In...	koleksinya lengkap dari seluruh provinsi di in...	koleksinya lengkap dari seluruh provinsi di in...	[koleksinya, lengkap, dari, seluruh, provinsi,...

PROSES STOPWORD REMOVAL

```
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

```
def remove_stopwords(text):
    return [word for word in text if word not in stop_words]

# Ubah hasil list jadi string
df['stopword removal'] = df['tokenize'].apply(
    lambda x: " ".join(remove_stopwords(x))
)

df.head(5)
```

	text	cleaning	case_folding	normalisasi	tokenize	stopword removal
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	[akhirnya, bisa, ikut, berpartisipasi, untuk, ...]	berpartisipasi kelengkapan fasilitas gedung si...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	[panas, gersang, sejak, pepohonan, ditebangin]	panas gersang pepohonan ditebangin
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	[sangat, bagus, karena, banyak, edukasi, untuk...]	bagus edukasi anak rekreasi plus edukasi sedih...
3	Koleksinya lengkap dari seluruh Provinsi	Koleksinya lengkap dari seluruh Provinsi	koleksinya lengkap dari seluruh provinsi di	koleksinya lengkap dari seluruh provinsi	[koleksinya, lengkap, dari, seluruh,	koleksinya lengkap provinsi indonesia

```
from collections import Counter

all_text = ' '.join(df['stopword removal'].astype(str))
all_text = re.sub(r'^a-zA-Z\s', '', all_text.lower())

word_list = all_text.split()
word_freq = Counter(word_list)

sorted_freq = sorted(word_freq.items(), key=lambda x: x[1], reverse=True)
df_freq = pd.DataFrame(sorted_freq, columns=['Kata', 'Frekuensi'])

df_freq.to_csv('frekuensi_kata_awal.csv', index=False)
print("✅ File 'frekuensi_kata_awal.csv' berhasil disimpan.")
df_freq.info()
df_freq.head(50)
```


✓ File 'frekuensi_kata_awal.csv' berhasil disimpan.
 <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 3169 entries, 0 to 3168
 Data columns (total 2 columns):
 # Column Non-Null Count Dtype

 0 Kata 3169 non-null object
 1 Frekuensi 3169 non-null int64
 dtypes: int64(1), object(1)
 memory usage: 49.6+ KB

Kata Frekuensi

```
hapus_kata = ['ya','ok','lon','the','and','of','a','i','with','this','for','its','you','is','sih']
```

```
# hapus kata dari teks
df['stopword removal'] = df['stopword removal'].apply(
    lambda x: ' '.join([kata for kata in x.split() if kata not in hapus_kata]))
```

```
df['stopword removal'].str.contains(' '.join(hapus_kata)).sum()
```

```
np.int64(1240)
```

```
6 tiket 162
```

```
from collections import Counter
import re
```

```
# --- Membuat kamus kata tidak baku -> kata baku
```

```
kamus_tidak_baku = {
    'bgs':'bagus',
    'bagu':'bagus'
}
```

```
def perbaiki_kata(teks):
    kata_list = teks.split()
    hasil = [kamus_tidak_baku.get(kata, kata) for kata in kata_list]
    return ' '.join(hasil)
```

```
df['stopword removal'] = df['stopword removal'].astype(str).apply(perbaiki_kata)
```

```
all_text = ' '.join(df['stopword removal'].astype(str))
all_text = re.sub(r'^a-zA-Z\s', '', all_text.lower())
```

```
word_list = all_text.split()
word_freq = Counter(word_list)
```

```
sorted_freq = sorted(word_freq.items(), key=lambda x: x[1], reverse=True)
df_freq = pd.DataFrame(sorted_freq, columns=['Kata', 'Frekuensi'])
```

```
# df_freq.info()
df_freq.head(50)
```

24	imersifa	71
25	harga	68
26	lantai	65
27	berkunjung	61
28	edukasi	59
29	orang	59
30	wisata	58
31	bersejarah	58
32	keluarga	54
33	menambah	54
34	museumnya	53
35	benda	53
36	pengetahuan	53
37	jam	52
38	pameran	51
39	peninggalan	51
40	ribu	51
41	anakanak	50
42	salah	48

43	online	48
44	koleksinya	47
45	the	47
46	murah	46
47	dewasa	46
48	jalan	46
49	beli	45

Kata Frekuensi

WORDCLOUD SETELAH PREPROCESSING

```

import pandas as pd
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# Buat stopwords tambahan
stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp', 'lu', 'deh', 'fyp', 'ya', 'gue', 'cs', 'kayak', 'nih', 'termul',
                  'sih', 'yg', 'nya', 'aja', 'sdh', 'gak', 'ga', 'update', 'apk', 'ajar', 'si', 'ok', 'gas', 'omon', 'nyinyi',
                  'no', 'rocky', 'party', 'kapolri', 'tni', 'panglima'])

# WordCloud Before
text_before = ' '.join(
    df['text']
    .astype(str)
    .tolist()
)

wc_before = WordCloud(
    stopwords=stopwords,
    background_color="white",
    max_words=500,
    width=800,
    height=400
).generate(text_before)

# WordCloud After
text_after = ' '.join(
    [" ".join(eval(x) if isinstance(x, str) and x.startswith('(') else x.split())
     for x in df['stopword removal'].astype(str)]
)

wc_after = WordCloud(
    stopwords=stopwords,
    background_color="white",
    max_words=500,
    width=800,
    height=400
).generate(text_after)

# Visualisasi side-by-side
plt.figure(figsize=(16, 8))

plt.subplot(1, 2, 1)
plt.imshow(wc_before, interpolation='bilinear')
plt.axis("off")
plt.title("Before Preprocessing", fontsize=18)

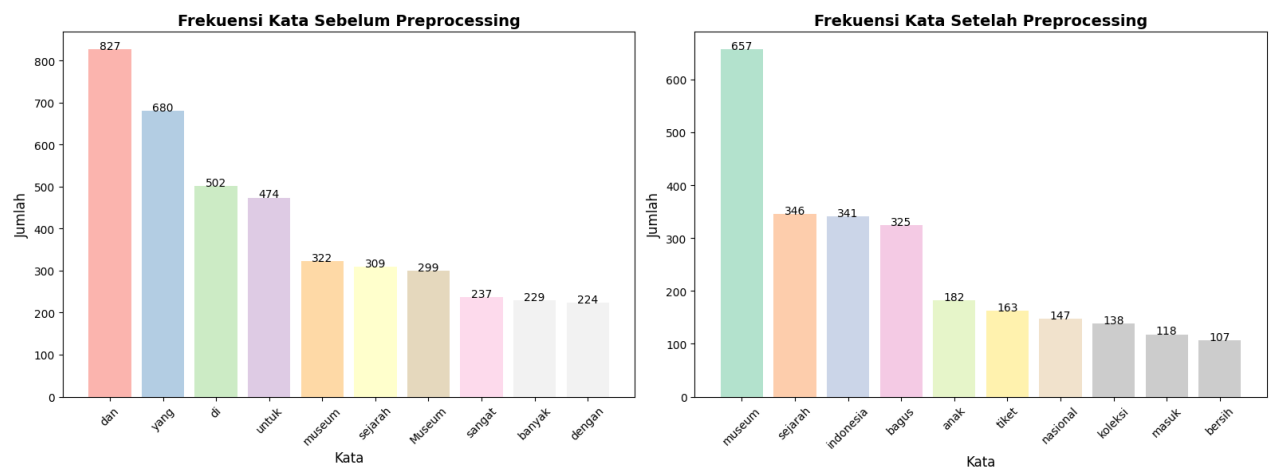
plt.subplot(1, 2, 2)
plt.imshow(wc_after, interpolation='bilinear')
plt.axis("off")
plt.title("After Preprocessing", fontsize=18)

plt.tight_layout()
plt.show()

```

37	pameran	51
38	peninggalan	51
39	ribu	51
40	anakanak	50
41	salah	48
42	online	48
43	koleksinya	47
44	murah	46
45	dewasa	46
46	jalan	46
47	beli	45
48	suka	45





```
df.to_csv('Hasil_Preprocessing_Data.csv',encoding='utf8', index=False);
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1280 entries, 0 to 1333
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   text                   1280 non-null   object
1   cleaning               1280 non-null   object
2   case_folding           1280 non-null   object
3   normalisasi            1280 non-null   object
4   tokenize               1280 non-null   object
5   stopword removal       1280 non-null   object
dtypes: object(6)
memory usage: 70.0+ KB
```

✓ **PELEBALAN DATA METODE LAXICON BASED 2 CLASS**

```
import pandas as pd

data = pd.read_csv("Hasil_Preprocessing_Data.csv")
data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1280 entries, 0 to 1279
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   text                   1280 non-null   object
1   cleaning               1279 non-null   object
2   case_folding           1279 non-null   object
3   normalisasi            1278 non-null   object
4   tokenize               1280 non-null   object
5   stopword removal       1269 non-null   object
dtypes: object(6)
memory usage: 60.1+ KB
```

	text	cleaning	case_folding	normalisasi	tokenize	stopword removal
0	Akhirnya bisa ikut berpartisipasi untuk keleng...	Akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	akhirnya bisa ikut berpartisipasi untuk keleng...	['akhirnya', 'bisa', 'ikut', 'berpartisipasi', ...]	berpartisipasi kelengkapan fasilitas gedung en...
1	Panas. Gersang sejak pepohonan ditebangin.	Panas Gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	panas gersang sejak pepohonan ditebangin	['panas', 'gersang', 'sejak', 'pepohonan', 'di...]	panas gersang pepohonan ditebangin
2	Sangat bagus karena banyak edukasi untuk anak2...	Sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	sangat bagus karena banyak edukasi untuk anak ...	['sangat', 'bagus', 'karena', 'banyak', 'eduka...]	bagus edukasi anak rekreasi plus edukasi sedih...
3	Koleksinya lengkap dari seluruh Provinsi di In...	Koleksinya lengkap dari seluruh Provinsi di In...	koleksinya lengkap dari seluruh provinsi di in...	koleksinya lengkap dari seluruh provinsi di in...	['koleksinya', 'lengkap', 'dari', 'seluruh', '...]	koleksinya lengkap provinsi indonesia ruangan ...
4	Pas akhir pekan jadi ramai dengan penquniung v...	Pas akhir pekan jadi ramai dengan penquniung v...	pas akhir pekan jadi ramai dengan penquniung v...	pas akhir pekan jadi ramai dengan penquniung v...	['pas', 'akhir', 'pekan', 'jadi', 'ramai', 'de...]	pas pekan ramai pengunjung kebanyakan

```
data = data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1269 entries, 0 to 1279
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   text                   1269 non-null   object
1   cleaning               1269 non-null   object
2   case_folding           1269 non-null   object
3   normalisasi            1269 non-null   object
4   tokenize               1269 non-null   object
5   stopword removal       1269 non-null   object
dtypes: object(6)
memory usage: 69.4+ KB
```

```
data = pd.DataFrame(data[['stopword removal']])
data.head(5)
```

	stopword removal
0	berpartisipasi kelengkapan fasilitas gedung en...
1	panas gersang pepohonan ditebangin
2	bagus edukasi anak rekreasi plus edukasi sedih...
3	koleksinya lengkap provinsi indonesia ruangan ...
4	pas pekan ramai pengunjung kebanyakan keluarga...

```
import pandas as pd
import requests
```

```
# Unduh kamus leksikon positif dan negatif dari GitHub
positive_url = "https://raw.githubusercontent.com/fajri91/InSet/master/positive.tsv"
negative_url = "https://raw.githubusercontent.com/fajri91/InSet/master/negative.tsv"
```

```
positive_lexicon = set(pd.read_csv(positive_url, sep="\t", header=None)[0])
negative_lexicon = set(pd.read_csv(negative_url, sep="\t", header=None)[0])
```

```
def determine_sentiment(text):
    if isinstance(text, str):
        positive_count = sum(1 for word in text.split() if word in positive_lexicon)
        negative_count = sum(1 for word in text.split() if word in negative_lexicon)
        sentiment_score = positive_count - negative_count
        sentiment = "Negatif" if sentiment_score <= -1 else "Positif"
        return sentiment_score, sentiment
    return -1, "Negatif"
```

```
data[['Score', 'Sentiment']] = data['stopword removal'].apply(lambda x: pd.Series(determine_sentiment(x)))
data.head(20)
```

	stopword removal	Score	Sentiment
0	berpartisipasi kelengkapan fasilitas gedung en...	1	Positif
1	panas gersang pepohonan ditebangin	-1	Negatif
2	bagus edukasi anak rekreasi plus edukasi sedih...	-2	Negatif
3	koleksinya lengkap provinsi indonesia ruangan ...	1	Positif
4	pas pekan ramai pengunjung kebanyakan keluarga...	2	Positif
5	kesini coba memakai guide gratis disediakan ti...	5	Positif
6	bagus wisata edukasi	0	Positif
7	menarik koleksi menampilkan kekayaan budaya ne...	3	Positif
8	menarik	0	Positif
9	mengunjungi museum bosan museumnya luas koleks...	2	Positif
10	menyenangkan berkunjung museum nasional penjel...	3	Positif
11	kemarin kesini pas pameran korea imersif seru ...	1	Positif
12	kesini pemeran keramik tempatnya oke banget co...	2	Positif
13	museum pola sirkulasi ramah anak berdua anak s...	2	Positif
14	jogging favorit event menarik	0	Positif
15	kesini akses transportasi mudah koleksi makan	2	Positif
16	museum gedung arcaarca panas banget museum ged...	-4	Negatif
17	mudah djangkau jenis moda transportasi amazing...	0	Positif
18	salah terbaik jakarta kunjungi menawarkan buda...	1	Positif
19	wajib kesini	-1	Negatif

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

# Menghitung jumlah sentimen

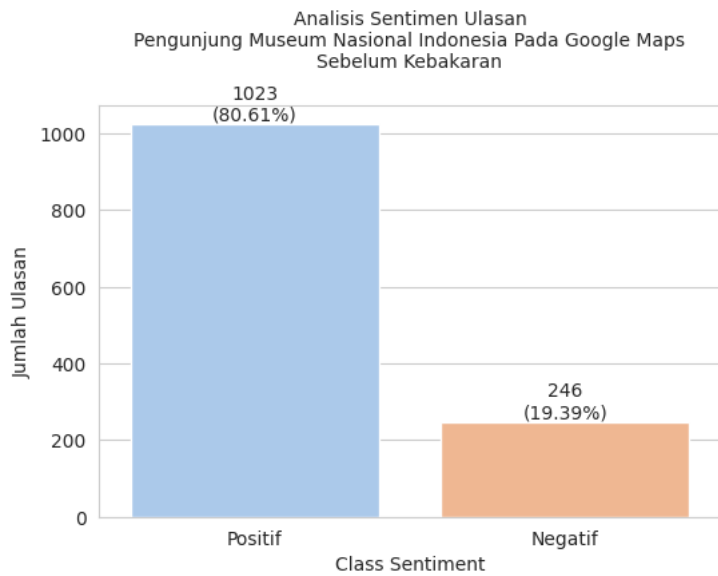
sentiment_count = data['Sentiment'].value_counts()
sns.set_style('whitegrid')

fig, ax = plt.subplots(figsize=(6, 4))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette='pastel')
plt.title('Analisis Sentimen Ulasan\nPengunjung Museum Nasional Indonesia Pada Google Maps\nSebelum Kebakaran', fontsize=16)
plt.xlabel('Class Sentiment', fontsize=10)
plt.ylabel('Jumlah Ulasan', fontsize=10)

total = len(data['Sentiment'])

for i, count in enumerate(sentiment_count.values):
    percentage = f'{100 * count / total:.2f}%'
    ax.text(i, count + 0.10, f'{count}\n({percentage})', ha='center', va='bottom')

plt.show()
```



```
data.to_csv('Hasil_Labelling_Data_Inset_Lax.csv',encoding='utf8', index=False)
```

✓ **Aspect Based Sentiment Classification (ABSC)**

```
import pandas as pd
```

```
data = pd.read_csv("Hasil_Labelling_Data_Inset_Lax.csv")
data.info()
data.head(5)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1269 entries, 0 to 1268
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   stopword removal      1269 non-null   object
1   Score                  1269 non-null   int64
2   Sentiment              1269 non-null   object
dtypes: int64(1), object(2)
memory usage: 29.9+ KB
```

	stopword removal	Score	Sentiment
0	berpartisipasi kelengkapan fasilitas gedung en...	1	Positif
1	panas gersang pepohonan ditebangin	-1	Negatif
2	bagus edukasi anak rekreasi plus edukasi sedih...	-2	Negatif
3	koleksinya lengkap provinsi indonesia ruangan ...	1	Positif
4	pas pekan ramai pengunjung kebanyakan keluarga...	2	Positif

```
data = data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1269 entries, 0 to 1268
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   stopword removal      1269 non-null   object
1   Score                  1269 non-null   int64
2   Sentiment              1269 non-null   object
dtypes: int64(1), object(2)
memory usage: 29.9+ KB
```

```
data = pd.DataFrame(data[['stopword removal', 'Sentiment']])
data.head(5)
```

	stopword removal	Sentiment
0	berpartisipasi kelengkapan fasilitas gedung en...	Positif
1	panas gersang pepohonan ditebangin	Negatif
2	bagus edukasi anak rekreasi plus edukasi sedih...	Negatif
3	koleksinya lengkap provinsi indonesia ruangan ...	Positif
4	pas pekan ramai pengunjung kebanyakan keluarga...	Positif

```

aspect_keywords = {
  "Aksesibilitas\n(Accesibility)": [
    # ===== 1 KATA =====
    "akses", "jalan", "transportasi", "mudah", "dekat",
    "strategis", "pusat", "kota", "halte", "stasiun",
    "mrt", "busway", "parkir", "lancar", "macet",

    # ===== 2 KATA =====
    "lokasi strategis", "pusat kota", "transportasi umum",
    "mudah dijangkau", "dekat halte", "dekat stasiun",
    "akses transportasi", "parkir mobil", "parkir motor",
    "jalan besar", "jalan utama",

    # ===== 3 KATA =====
    "lokasi di pusat kota", "mudah dijangkau transportasi",
    "dekat transportasi umum", "akses transportasi mudah",
    "transportasi umum tersedia", "akses menuju museum",
    "jalan menuju museum"
  ],

  "Pelayanan\n(Service)": [
    # ===== 1 KATA =====
    "ramah", "judes", "profesional", "pemandu", "guide", "servis",
    "service", "etika", "sopan", "responsif", "membantu", "aktif", "cepat", "kasar",
    "kaku", "bingung", "berbelit", "acuh", "even", "event", "senang", "menyenangkan",

    # ===== 2 KATA =====
    "tidak ramah", "gag ramah", "ga ramah", "tidak profesional", "antri lama",
    "guide informatif", "petunjuk minim", "kurang membantu", "tidak jelas",
    "penjelasan lengkap", "respon baik", "tersedia pemandu",
    "tersedia guide", "guide gratis",
  ],

  "Fasilitas\n(Amenities)": [
    # ===== 1 KATA =====
    "toilet", "parkir", "wifi", "mushola", "gedung", "lobby",
    "ruang", "galeri", "lift", "eskalator", "ac",
    "pendingin", "loker", "tiket", "loket", "mushola", "kantin",
    "cafe", "petugas", "keamanan", "kebersihan", "panas",
    "gersang", "pintu", "penerangan", "kursi", "jalur", "alur", "denah", "antrian",
    "seran", "difabel", "bersih", "nyaman", "lengkap", "modern", "sempit",
    "sepi", "ribet", "lambat", "musholla",

    # ===== 2 KATA =====
    "toilet umum", "area parkir", "tempat parkir", "parkir banjir"
    "ruang pameran", "gedung museum", "loket tiket",
    "tiket masuk", "petugas ramah", "petugas informatif",
    "audio guide", "ruang edukasi", "fasilitas museum", "ac mati",
    "toilet kotor", "ruangan panas", "parkiran sempit", "antrian panjang", "antrian lama",
    "tempat duduk"

    # ===== 3 KATA =====
    "fasilitas museum lengkap", "ruang pameran luas",
    "gedung museum megah", "toilet bersih tersedia",
    "petugas sangat ramah", "sistem tiket rapi",
    "fasilitas edukasi lengkap"
  ],

  "Koleksi\n(collection)": [
    # ===== 1 KATA =====
    "sejarah", "bersejarah", "edukatif", "informatif", "menarik",
    "unik", "ikonik", "megah", "luas", "klasik",
    "budaya", "koleksi", "artefak",
    "estetik", "legendaris", "imersif", "rapi", "terorganisir", "pameran",
    "organisir", "bagus", "arca", "lukisan", "keris", "rusak", "indah", "artefak",
    "koleksi", "edukasi", "membosankan", "minim", "biasa", "beragam",

    # ===== 2 KATA =====
    "museum sejarah", "wisata edukasi",
    "wisata sejarah", "koleksi sejarah", "koleksi budaya",
    "benda bersejarah", "nilai sejarah", "galeri museum",
  ]
}

```

```

    "pameran tetap", "pameran temporer", "objek bersejarah",
    "tata letak", "ruang imersifa", "ruang imersif", "isi sedikit",

    # ===== 3 KATA =====
    "koleksi sejarah lengkap",
    "wisata edukasi sejarah", "benda peninggalan sejarah",
    "koleksi budaya indonesia", "pameran sejarah menarik",
    "museum sejarah nasional", "koleksi museum lengkap",
]
}

# daftar semua aspek yang ada
all_aspects = list(aspect_keywords.keys())
counter = 0 # untuk round robin

def aspect_extraction(text):
    global counter
    for aspect, keywords in aspect_keywords.items():
        if any(keyword in text for keyword in keywords):
            return aspect
    # jika tidak ketemu keyword -> dibagi rata
    aspect = all_aspects[counter % len(all_aspects)]
    counter += 1
    return aspect

# terapkan ke dataframe
data['aspect_extraction'] = data['stopword removal'].astype(str).apply(aspect_extraction)
data.head(20)

```

	stopword removal	Sentiment	aspect_extraction
0	berpartisipasi kelengkapan fasilitas gedung en...	Positif	Fasilitas\n(Amenities)
1	panas gersang pepohonan ditebangin	Negatif	Fasilitas\n(Amenities)
2	bagus edukasi anak rekreasi plus edukasi sedih...	Negatif	Koleksi\n(collection)
3	koleksinya lengkap provinsi indonesia ruangan ...	Positif	Fasilitas\n(Amenities)
4	pas pekan ramai pengunjung kebanyakan keluarga...	Positif	Fasilitas\n(Amenities)
5	kesini coba memakai guide gratis disediakan ti...	Positif	Pelayanan\n(Service)
6	bagus wisata edukasi	Positif	Koleksi\n(collection)
7	menarik koleksi menampilkan kekayaan budaya ne...	Positif	Fasilitas\n(Amenities)
8	menarik	Positif	Koleksi\n(collection)
9	mengunjungi museum bosan museumnya luas koleks...	Positif	Pelayanan\n(Service)
10	menyenangkan berkunjung museum nasional penjel...	Positif	Aksesibilitas\n(Accesibility)
11	kemarin kesini pas pameran korea imersif seru ...	Positif	Fasilitas\n(Amenities)
12	kesini pemeran keramik tempatnya oke banget co...	Positif	Aksesibilitas\n(Accesibility)
13	museum pola sirkulasi ramah anak berdua anak s...	Positif	Pelayanan\n(Service)
14	jogging favorit event menarik	Positif	Pelayanan\n(Service)
15	kesini akses transportasi mudah koleksi makan	Positif	Aksesibilitas\n(Accesibility)
16	museum gedung arcaarca panas banget museum ged...	Negatif	Fasilitas\n(Amenities)
17	mudah djangkau jenis moda transportasi amazing...	Positif	Aksesibilitas\n(Accesibility)
18	salah terbaik jakarta kunjungi menawarkan buda...	Positif	Koleksi\n(collection)
19	wajib kesini	Negatif	Aksesibilitas\n(Accesibility)

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

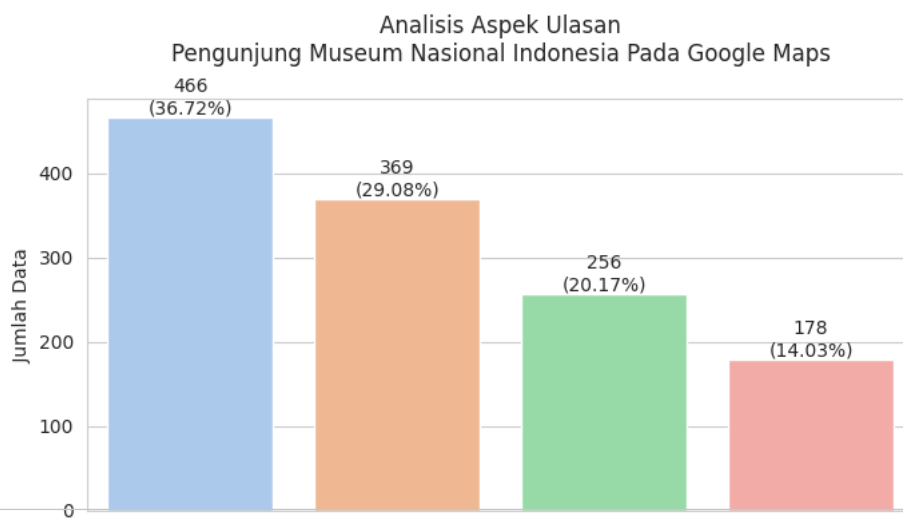
sentiment_count = data['aspect_extraction'].value_counts()
sns.set_style('whitegrid')

fig, ax = plt.subplots(figsize=(8, 4))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette='pastel')
plt.title('Analisis Aspek Ulasan\nPengunjung Museum Nasional Indonesia Pada Google Maps', fontsize=12, pad=20)
plt.xlabel(' Class Aspek ', fontsize=10)
plt.ylabel(' Jumlah Data ', fontsize=10)

total = len(data['aspect_extraction'])

for i, count in enumerate(sentiment_count.values):
    percentage = f'{100 * count / total:.2f}%'
    ax.text(i, count + 0.10, f'{count}\n({percentage})', ha='center', va='bottom')

plt.show()
```



```
data.to_csv('Hasil_AspectExtraction_Data.csv',encoding='utf8', index=False)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1269 entries, 0 to 1268
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   stopword removal      1269 non-null   object
1   Sentiment              1269 non-null   object
2   aspect_extraction     1269 non-null   object
dtypes: object(3)
memory usage: 29.9+ KB
```

Start coding or [generate](#) with AI.

✓ WORDCLOUD & FREKUENSI KATA NGRAM

```
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import CountVectorizer
from wordcloud import WordCloud

# Ganti dengan path dataset kamu
df = pd.read_csv("Hasil_AspectExtraction_Data.csv")

df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1269 entries, 0 to 1268
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   stopword removal      1269 non-null   object
1   Sentiment              1269 non-null   object
2   aspect_extraction      1269 non-null   object
dtypes: object(3)
memory usage: 29.9+ KB
```

	stopword removal	Sentiment	aspect_extraction
0	berpartisipasi kelengkapan fasilitas gedung en...	Positif	Fasilitas\n(Amenities)
1	panas gersang pepohonan ditebangin	Negatif	Fasilitas\n(Amenities)
2	bagus edukasi anak rekreasi plus edukasi sedih...	Negatif	Koleksi\n(collection)
3	koleksinya lengkap provinsi indonesia ruangan ...	Positif	Fasilitas\n(Amenities)
4	pas pekan ramai pengunjung kebanyakan keluarga...	Positif	Fasilitas\n(Amenities)

```
# Pastikan tidak ada nilai kosong
df = df.dropna(subset=['stopword removal', 'Sentiment'])
df.head()
```

	stopword removal	Sentiment	aspect_extraction
0	berpartisipasi kelengkapan fasilitas gedung en...	Positif	Fasilitas\n(Amenities)
1	panas gersang pepohonan ditebangin	Negatif	Fasilitas\n(Amenities)
2	bagus edukasi anak rekreasi plus edukasi sedih...	Negatif	Koleksi\n(collection)
3	koleksinya lengkap provinsi indonesia ruangan ...	Positif	Fasilitas\n(Amenities)
4	pas pekan ramai pengunjung kebanyakan keluarga...	Positif	Fasilitas\n(Amenities)

```
def wordcloud_ngram(data, sentiment_label, ngram_range=(2,3), max_words=1000):
    text_data = data[data['Sentiment'] == sentiment_label]['stopword removal']
    text_data = text_data.dropna()
    text_data = text_data[text_data.str.len() > 0]

    vectorizer = CountVectorizer(ngram_range=ngram_range)
    X = vectorizer.fit_transform(text_data)

    word_freq = dict(zip(
        vectorizer.get_feature_names_out(),
        X.sum(axis=0).A1
    ))

    # filter frekuensi kecil
    word_freq = {k: v for k, v in word_freq.items() if v > 2}

    wc = WordCloud(
        width=1000,
        height=500,
        background_color='white',
        max_words=max_words,
        prefer_horizontal=0.9,
        relative_scaling=0.5,
        collocations=False,
        min_font_size=8,
        max_font_size=120,
        scale=5
    ).generate_from_frequencies(word_freq)

    plt.figure(figsize=(8,4))
    plt.imshow(wc, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'WordCloud {ngram_range[0]}-gram Sentimen {sentiment_label}', fontsize=14)
    plt.show()

# Bigram
wordcloud_ngram(df, 'Positif', ngram_range=(2,2))
print()
wordcloud_ngram(df, 'Negatif', ngram_range=(2,2))
```

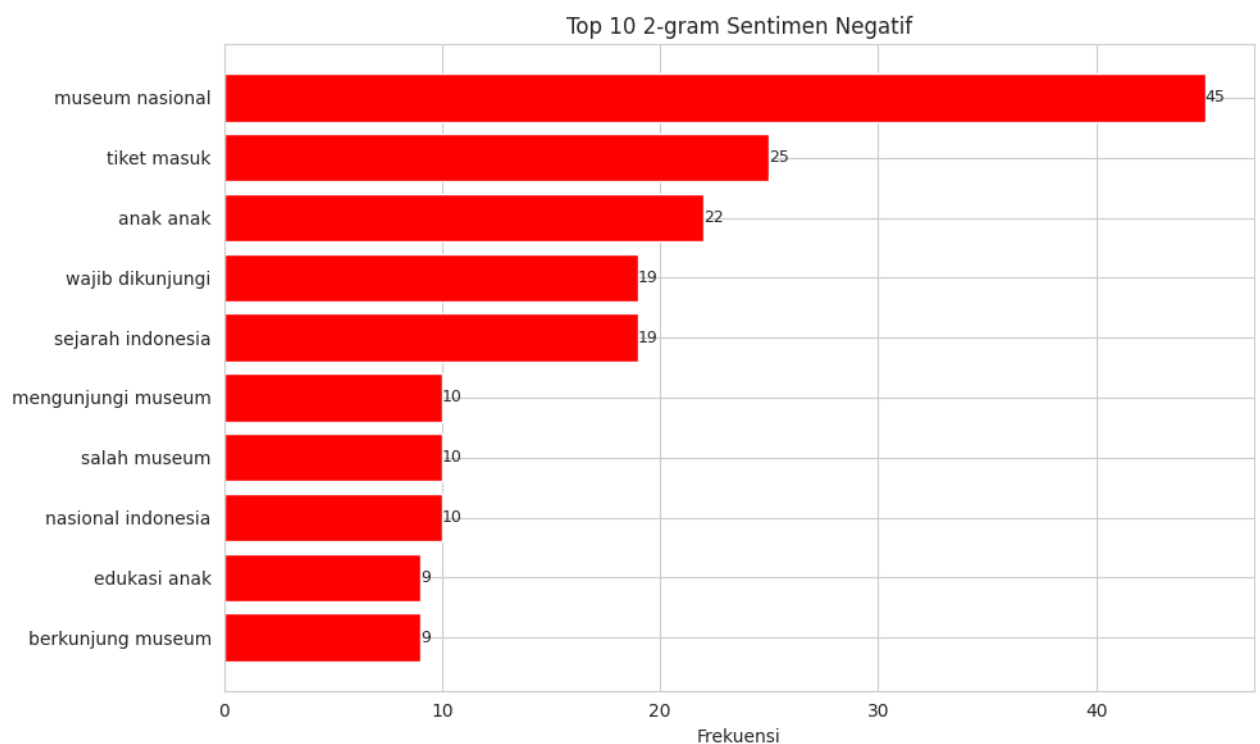
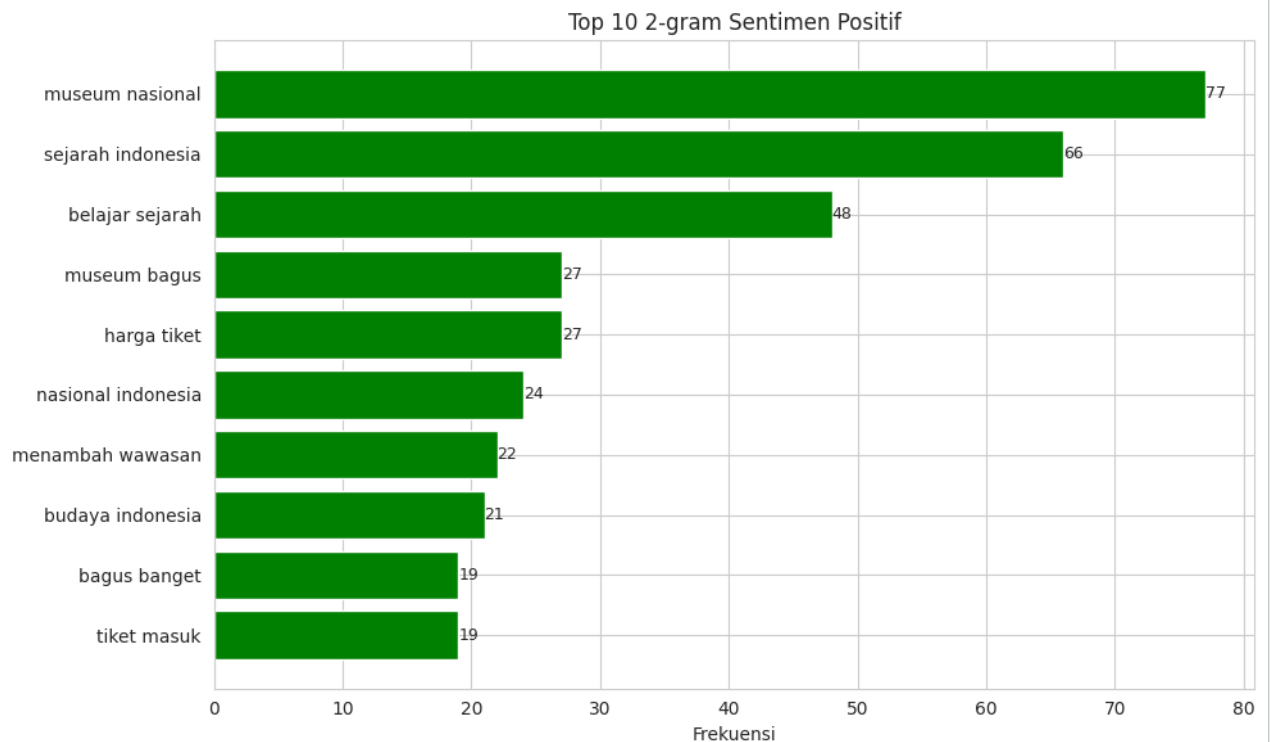


19/22

```

plot_top_ngram(df, 'Positif', ngram_range=(2,2), color='green')
print()
plot_top_ngram(df, 'Negatif', ngram_range=(2,2), color='red')

```



PENERAPAN ALGORITMA SVM - Karna Linear

```

import pandas as pd

file_path = 'Hasil_AspectExtraction_Data.csv'
data = pd.read_csv(file_path)

data.info()
data.head()

```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1269 entries, 0 to 1268
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   stopword removal      1269 non-null   object
1   Sentiment              1269 non-null   object
2   aspect_extraction      1269 non-null   object
dtypes: object(3)
memory usage: 29.9+ KB
```

	stopword removal	Sentiment	aspect_extraction
0	berpartisipasi kelengkapan fasilitas gedung en...	Positif	Fasilitas\n(Amenities)
1	panas gersang pepohonan ditebangin	Negatif	Fasilitas\n(Amenities)
2	bagus edukasi anak rekreasi plus edukasi sedih...	Negatif	Koleksi\n(collection)
3	koleksinya lengkap provinsi indonesia ruangan ...	Positif	Fasilitas\n(Amenities)
4	pas pekan ramai pengunjung kebanyakan keluarga...	Positif	Fasilitas\n(Amenities)

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
```

```
cleaned_data = data.dropna(subset=['stopword removal'])
```

```
X = cleaned_data['stopword removal']
y = cleaned_data['Sentiment']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

```
print("Jumlah data latih:", len(X_train))
print("Jumlah data uji:", len(X_test))
print('=====')
```

```
Jumlah data latih: 1015
Jumlah data uji: 254
=====
```

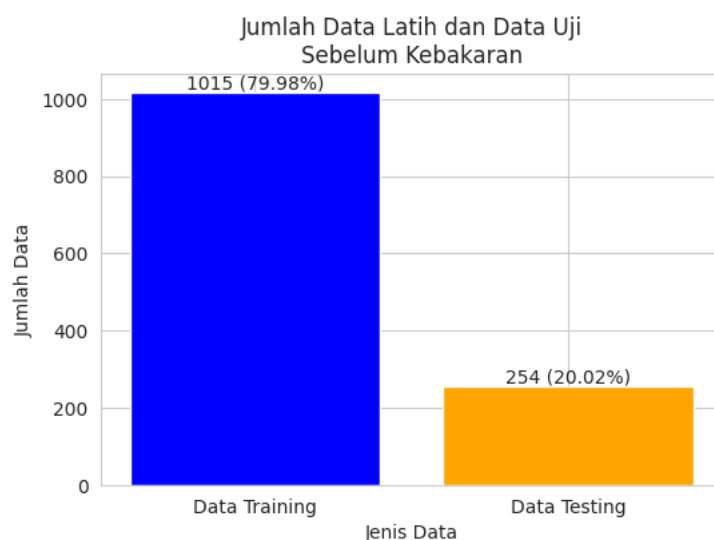
```
import matplotlib.pyplot as plt
```

```
train_size = len(X_train)
test_size = len(X_test)
```

```
plt.figure(figsize=(6, 4))
bars = plt.bar(['Data Training', 'Data Testing'], [train_size, test_size], color=['blue', 'orange'])
```

```
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height} ({height / (train_size + test_size) * 100:.2f}%)',
             ha='center', va='bottom')
```

```
plt.title('Jumlah Data Latih dan Data Uji\nSebelum Kebakaran')
plt.xlabel('Jenis Data')
plt.ylabel('Jumlah Data')
plt.show()
```



```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

# Inisialisasi TF-IDF Vectorizer
vectorizer = TfidfVectorizer()

# Fit & transform data train
X_train_vec = vectorizer.fit_transform(X_train)

# Transform data test (tanpa fit ulang)
X_test_vec = vectorizer.transform(X_test)

# Cek shape
print("Shape of X_train_vec:", X_train_vec.shape)
print("Shape of X_test_vec:", X_test_vec.shape)

# Lihat hasil vektorisasi (array)
print("\nX_train_vec (as array):")
print(X_train_vec.toarray())

print("\nX_test_vec (as array):")
print(X_test_vec.toarray())

# Konversi ke DataFrame
df_X_train_vec = pd.DataFrame(
    X_train_vec.toarray(),
    columns=vectorizer.get_feature_names_out()
```

TEXT PREPROCESSING

PROSES CLEANING

```
import pandas as pd
```

```
df = pd.read_csv("Periode Setelah.csv")
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1617 entries, 0 to 1616
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0    text    1617 non-null    object
dtypes: object(1)
memory usage: 12.8+ KB
```

text

```
0    kebetulan saya ke monas pas hari HUT TNI jadi ...
1    Banyak arca dan cukup menarik untuk anak2 teru...
2    sangatt seru sekali liburan bersama keluarga k...
3    Pengalaman yang menyenangkan
4    Rekomendasi tempat alternatif liburan sekaligu...
```

```
df = pd.DataFrame(df[['text']])
df.info()
df.head(5)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1617 entries, 0 to 1616
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0    text    1617 non-null    object
dtypes: object(1)
memory usage: 12.8+ KB
```

text

```
0    kebetulan saya ke monas pas hari HUT TNI jadi ...
1    Banyak arca dan cukup menarik untuk anak2 teru...
2    sangatt seru sekali liburan bersama keluarga k...
3    Pengalaman yang menyenangkan
4    Rekomendasi tempat alternatif liburan sekaligu...
```

```
#menghapus data duplikat
```

```
df.drop_duplicates(subset ="text", inplace = True)
```

```
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1584 entries, 0 to 1616
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0    text    1584 non-null    object
dtypes: object(1)
memory usage: 24.8+ KB
```

text

```
0    kebetulan saya ke monas pas hari HUT TNI jadi ...
1    Banyak arca dan cukup menarik untuk anak2 teru...
2    sangatt seru sekali liburan bersama keluarga k...
3    Pengalaman yang menyenangkan
4    Rekomendasi tempat alternatif liburan sekaligu...
```

```

import re
import string
import nltk

# Fungsi untuk menghapus URL
def remove_URL(tweet):
    if tweet is not None and isinstance(tweet, str):
        url = re.compile(r'https?://\S+|www\.\S+')
        return url.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus HTML
def remove_html(tweet):
    if tweet is not None and isinstance(tweet, str):
        html = re.compile(r'<.*?>')
        return html.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus emoji
def remove_emoji(tweet):
    if tweet is not None and isinstance(tweet, str):
        emoji_pattern = re.compile("["
            u"\U0001F600-\U0001F64F" # emoticons
            u"\U0001F300-\U0001F5FF" # symbols & pictographs
            u"\U0001F680-\U0001F6FF" # transport & map symbols
            u"\U0001F700-\U0001F77F" # alchemical symbols
            u"\U0001F780-\U0001F7FF" # Geometric Shapes Extended
            u"\U0001F800-\U0001F8FF" # Supplemental Arrows-C
            u"\U0001F900-\U0001F9FF" # Supplemental Symbols and Pictographs
            u"\U0001FA00-\U0001FA6F" # Chess Symbols
            u"\U0001FA70-\U0001FAFF" # Symbols and Pictographs Extended-A
            u"\U0001F004-\U0001F0CF" # Additional emoticons
            u"\U0001F1E0-\U0001F1FF" # flags
            "]+", flags=re.UNICODE)
        return emoji_pattern.sub(r'', tweet)
    else:
        return tweet

# Fungsi untuk menghapus simbol
def remove_symbols(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'^a-zA-Z0-9\s', '', tweet)
        return tweet

# Fungsi untuk menghapus angka
def remove_numbers(tweet):
    if tweet is not None and isinstance(tweet, str):
        tweet = re.sub(r'\d', '', tweet)
        return tweet

# Fungsi hapus username
def remove_usernames(text):
    if text is not None and isinstance(text, str):
        return re.sub(r'@\w+', '', text)
    else:
        return text

df['cleaning'] = df['text'].apply(lambda x: remove_URL(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_usernames(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))

df.head(5)

```

	text	cleaning
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...
3	Pengalaman yang menyenangkan	Pengalaman yang menyenangkan
4	Rekomendasi tempat alternatif liburan sekaligu...	Rekomendasi tempat alternatif liburan sekaligu...

PROSES CASE FOLDING

```
def case_folding(text):
    if isinstance(text, str):
        lowercase_text = text.lower()
        return lowercase_text
    else:
        return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(5)
```

	text	cleaning	case_folding
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...
3	Pengalaman vana menvenanakan	Pengalaman vana menvenanakan	pengalaman vana menvenanakan

Normalisasi Kata

```
import pandas as pd
import requests
from io import BytesIO

# Fungsi penggantian kata tidak baku
def replace_taboo_words(text, kamus_tidak_baku):
    if isinstance(text, str):
        words = text.split()
        replaced_words = []
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []

        for word in words:
            if word in kamus_tidak_baku:
                baku_word = kamus_tidak_baku[word]
                if isinstance(baku_word, str) and all(char.isalpha() for char in baku_word):
                    replaced_words.append(baku_word)
                    kalimat_baku.append(baku_word)
                    kata_diganti.append(word)
                    kata_tidak_baku_hash.append(hash(word))
            else:
                replaced_words.append(word)
        replaced_text = ' '.join(replaced_words)
    else:
        replaced_text = ''
        kalimat_baku = []
        kata_diganti = []
        kata_tidak_baku_hash = []

    return replaced_text, kalimat_baku, kata_diganti, kata_tidak_baku_hash

# Baca dataset kamu (pastikan df sudah tersedia)
data = pd.DataFrame(df[['text', 'cleaning', 'case_folding']])
data.head()
```

	text	cleaning	case_folding
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...
3	Pengalaman vana menyenangkan	Pengalaman vana menvenanakan	pengalaman vana menvenanakan

```
# Unduh dan baca kamus dari GitHub
url = "https://github.com/analysisdatasentiment/kamus_kata_baku/raw/main/kamuskatabaku.xlsx"
```

```
response = requests.get(url)
file_excel = BytesIO(response.content)
kamus_data = pd.read_excel(file_excel)

# Buat dictionary dari kamus
kamus_tidak_baku_dict = dict(zip(kamus_data['tidak_baku'], kamus_data['kata_baku']))
```

```
# Terapkan fungsi normalisasi
data[['normalisasi', 'Kata_Baku', 'Kata_Tidak_Baku', 'Kata_Tidak_Baku_Hash']] = data['case_folding'].apply(
    lambda x: pd.Series(replace_taboo_words(x, kamus_tidak_baku_dict))
)

# Ambil kolom yang relevan
df = pd.DataFrame(data[['text', 'cleaning', 'case_folding', 'normalisasi']])
df.head(20)
```

	text	cleaning	case_folding	normalisasi
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...
3	Pengalaman yang menyenangkan	Pengalaman yang menyenangkan	pengalaman yang menyenangkan	pengalaman yang menyenangkan
4	Rekomendasi tempat alternatif liburan sekaligu...	Rekomendasi tempat alternatif liburan sekaligu...	rekomendasi tempat alternatif liburan sekaligu...	rekomendasi tempat alternatif liburan sekaligu...
5	"- Museum Nasional Tanpa Anggaran -\n\nMenyebu...	Museum Nasional Tanpa Anggaran \n\nMenyebut t...	museum nasional tanpa anggaran \n\nmenyebut t...	museum nasional tanpa anggaran menyebut tempat...
6	"Museum yang bagus dengan koleksi dewa-dewa In...	Museum yang bagus dengan koleksi dewadewa Indo...	museum yang bagus dengan koleksi dewadewa indo...	museum yang bagus dengan koleksi dewadewa indo...
7	seru banget	seru banget	seru banget	seru banget
8	Sekali seumur hidup, datanglah kesini\nMengenal...	Sekali seumur hidup datanglah kesini\nMengenal...	sekali seumur hidup datanglah kesini\nmenengal...	sekali seumur hidup datanglah kesini mengenali...
9	Tempat bersih pelayanan bagus ,\nSecurity ramah...	Tempat bersih pelayanan bagus \nSecurity ramah...	tempat bersih pelayanan bagus \nsecurity ramah...	tempat bersih pelayanan bagus security ramah d...
10	Itu adalah pengalaman yang luar biasa bagi ana...	Itu adalah pengalaman yang luar biasa bagi ana...	itu adalah pengalaman yang luar biasa bagi ana...	itu adalah pengalaman yang luar biasa bagi ana...
11	🌟 Museum Nasional Indonesia terasa seperti mas...	Museum Nasional Indonesia terasa seperti masu...	museum nasional indonesia terasa seperti masu...	museum nasional indonesia terasa seperti masuk...
12	Kerja bagus	Kerja bagus	kerja bagus	kerja bagus
13	Museum yang bagus untuk lebih mengenal Sejarah...	Museum yang bagus untuk lebih mengenal Sejarah...	museum yang bagus untuk lebih mengenal sejarah...	museum yang bagus untuk lebih mengenal sejarah...
14	Terawat dengan baik dan sangat informatif. Ada...	Terawat dengan baik dan sangat informatif Ada ...	terawat dengan baik dan sangat informatif ada ...	terawat dengan baik dan sangat informatif ada ...
15	Museum yang sangat cocok untuk belajar sejarah...	Museum yang sangat cocok untuk belajar sejarah...	museum yang sangat cocok untuk belajar sejarah...	museum yang sangat cocok untuk belajar sejarah...
16	Dibutuhkan sekitar setengah jam untuk melihat ...	Dibutuhkan sekitar setengah jam untuk melihat ...	dibutuhkan sekitar setengah jam untuk melihat ...	dibutuhkan sekitar setengah jam untuk melihat ...
17	Semua yang dapat Anda temukan tentang sejarah ...	Semua yang dapat Anda temukan tentang sejarah ...	semua yang dapat anda temukan tentang sejarah ...	semua yang dapat anda temukan tentang sejarah ...
18	Anda bisa belajar banyak sejarah dari museum ini.	Anda bisa belajar banyak sejarah dari museum ini	anda bisa belajar banyak sejarah dari museum ini	anda bisa belajar banyak sejarah dari museum ini

TOKENIZATION

```
def tokenize(text):
    tokens = text.split()
    return tokens

df['tokenize'] = df['normalisasi'].apply(tokenize)

df.head(5)
```

	text	cleaning	case_folding	normalisasi	tokenize
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	[kebetulan, saya, ke, monas, pas, hari, hut, t...
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	[banyak, arca, dan, cukup, menarik, untuk, ana...
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	[sangatt, seru, sekali, liburan, bersama, kelu...
3	Pengalaman yang menyenangkan	Pengalaman yang menyenangkan	pengalaman yang menyenangkan	pengalaman yang menyenangkan	[pengalaman, yang, menyenangkan]
4	Rekomendasi tempat alternatif liburan sekaligu...	Rekomendasi tempat alternatif liburan sekaligu...	rekomenadasi tempat alternatif liburan sekaligu...	rekomenadasi tempat alternatif liburan sekaligu...	[rekomenadasi, tempat, alternatif, liburan, sek...

PROSES STOPWORD REMOVAL

```
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('indonesian')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
```

```
def remove_stopwords(text):
    return [word for word in text if word not in stop_words]

# Ubah hasil list jadi string
df['stopword removal'] = df['tokenize'].apply(
    lambda x: " ".join(remove_stopwords(x))
)

df.head(5)
```

	text	cleaning	case_folding	normalisasi	tokenize	stopword removal
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	[kebetulan, saya, ke, monas, pas, hari, hut, t...	monas pas hut tni padat masuk
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	[banyak, arca, dan, cukup, menarik, untuk, ana...	arca menarik anak ruangan imersifnya
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	[sangatt, seru, sekali, liburan, bersama, kelu...	sangatt seru liburan keluarga kesini benda ber...
3	Pengalaman yang menyenangkan	Pengalaman yang menyenangkan	pengalaman yang menyenangkan	pengalaman yang menyenangkan	[pengalaman, yang, menyenangkan]	pengalaman menyenangkan

```
from collections import Counter

all_text = ' '.join(df['stopword removal'].astype(str))
all_text = re.sub(r'^a-zA-Z\s', '', all_text.lower())

word_list = all_text.split()
word_freq = Counter(word_list)

sorted_freq = sorted(word_freq.items(), key=lambda x: x[1], reverse=True)
df_freq = pd.DataFrame(sorted_freq, columns=['Kata', 'Frekuensi'])

df_freq.to_csv('frekuensi_kata_awal.csv', index=False)
print("✅ File 'frekuensi_kata_awal.csv' berhasil disimpan.")
df_freq.info()
df_freq.head(50)
```

✓ File 'frekuensi_kata_awal.csv' berhasil disimpan.
 <class 'pandas.core.frame.DataFrame'>
 RangeIndex: 4852 entries, 0 to 4851
 Data columns (total 2 columns):
 # Column Non-Null Count Dtype

 0 Kata 4852 non-null object
 1 Frekuensi 4852 non-null int64
 dtypes: int64(1), object(1)
 memory usage: 75.9+ KB

	Kata	Frekuensi
0	museum	1038
1	tiket	471
2	indonesia	466
3	bagus	404
4	sejarah	404
5	ya	395
6	anak	331
7	nasional	292
8	masuk	291
9	banget	267
10	pameran	236
11	koleksi	225
12	imersifa	188
13	menarik	173
14	beli	173
15	ruang	169
16	lantai	148
17	luas	133
18	kesini	123
19	bersih	123
20	tempatnya	121
21	pengunjung	119
22	online	119
23	orang	116
24	nyaman	115

```
hapus_kata = ['ya','ok','lon','the','and','of','a','i','with','this','for','its','you','is','sih']
```

```
# hapus kata dari teks
df['stopword removal'] = df['stopword removal'].apply(
    lambda x: ' '.join([kata for kata in x.split() if kata not in hapus_kata]))

df['stopword removal'].str.contains(' '.join(hapus_kata)).sum()
```

```
0 int64(1557) 106
```

```
from collections import Counter
import re
```

```
# --- Membuat kamus kata tidak baku -> kata baku
```

```
kamus_tidak_baku = {
    'bgs':'bagus',
    'bagu':'bagus'
}
```

```
def perbaiki_kata(teks):
    kata_list = teks.split()
    hasil = [kamus_tidak_baku.get(kata, kata) for kata in kata_list]
    return ' '.join(hasil)
```

```
df['stopword removal'] = df['stopword removal'].astype(str).apply(perbaiki_kata)
```

```
all_text = ' '.join(df['stopword removal'].astype(str))
all_text = re.sub(r'[^a-zA-Z\s]', '', all_text.lower())
```

```
word_list = all_text.split()
word_freq = Counter(word_list)

sorted_freq = sorted(word_freq.items(), key=lambda x: x[1], reverse=True)
df_freq = pd.DataFrame(sorted_freq, columns=['Kata', 'Frekuensi'])

# df_freq.info()
df_freq.head(50)
```

49	traveloka	71
	Kata	Frekuensi
0	museum	1038
1	tiket	471
2	indonesia	466
3	bagus	404
4	sejarah	404
5	anak	331
6	nasional	292
7	masuk	291
8	banget	267
9	pameran	236
10	koleksi	225
11	imersifa	188
12	menarik	173
13	beli	173
14	ruang	169
15	lantai	148
16	luas	133
17	kesini	123
18	bersih	123
19	tempatnya	121
20	pengunjung	119
21	online	119
22	orang	116
23	nyaman	115
24	kebakaran	114
25	belajar	112
26	berkunjung	112
27	keren	112
28	seru	107
29	pas	106
30	arca	106

WORDCLOUD SETELAH PREPROCESSING

32	dikunjungi	102
----	------------	-----

```
import pandas as pd
import numpy as np
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
import matplotlib.pyplot as plt

# Buat stopwords tambahan
stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp', 'lu', 'deh', 'fyp', 'ya', 'gue', 'cs', 'kayak', 'nih', 'termul',
                  'sih', 'yg', 'nya', 'aja', 'sdh', 'gak', 'ga', 'update', 'apk', 'ajar', 'si', 'ok', 'gas', 'omon', 'nyinyi',
                  'no', 'rocky', 'party', 'kapolri', 'tni', 'panglima'])

# WordCloud Before
text_before = ' '.join(
    df['text']
    .astype(str)
```

```

        .tolist()
    )

wc_before = WordCloud(
    stopwords=stopwords,
    background_color="white",
    max_words=500,
    width=800,
    height=400
).generate(text_before)

# WordCloud After
text_after = ' '.join(
    [" ".join(eval(x) if isinstance(x, str) and x.startswith('(') else x.split())
     for x in df['stopword removal'].astype(str)]
)

wc_after = WordCloud(
    stopwords=stopwords,
    background_color="white",
    max_words=500,
    width=800,
    height=400
).generate(text_after)

# Visualisasi side-by-side
plt.figure(figsize=(16, 8))

plt.subplot(1, 2, 1)
plt.imshow(wc_before, interpolation='bilinear')
plt.axis("off")
plt.title("Before Preprocessing", fontsize=18)

plt.subplot(1, 2, 2)
plt.imshow(wc_after, interpolation='bilinear')
plt.axis("off")
plt.title("After Preprocessing", fontsize=18)

plt.tight_layout()
plt.show()

```



```

import matplotlib.pyplot as plt
from collections import Counter
from wordcloud import STOPWORDS

# Set stopwords tambahan
stopwords = set(STOPWORDS)
stopwords.update(['https', 'co', 'RT', '...', 'amp', 'lu', 'deh', 'fyp', 'ya', 'gue', 'cs', 'kayak', 'nih', 'termul',
                  'sih', 'yg', 'nya', 'aja', 'sdh', 'gak', 'ga', 'update', 'apk', 'ajar', 'si', 'ok', 'gas', 'omon', 'nyinyi',
                  'no', 'rocky', 'party', 'kapolri', 'tni', 'panglima', 'banget', 'diganti'])

# Fungsi helper untuk membersihkan tanda kutip, [ ], , dll
def clean_text_column(series):
    cleaned = []
    for text in series.astype(str):
        # Hapus [ ], tanda petik tunggal/ganda, koma
        text = re.sub(r"[\[\]'\"]", "", text)
        cleaned.append(text.strip())
    return " ".join(cleaned)

```

```
# BEFORE
all_text_before = clean_text_column(df["text"])
words_before = all_text_before.split()
filtered_words_before = [word for word in words_before if word.lower() not in stopwords]
word_counts_before = Counter(filtered_words_before)
top_words_before = word_counts_before.most_common(10)
word_before, count_before = zip(*top_words_before)

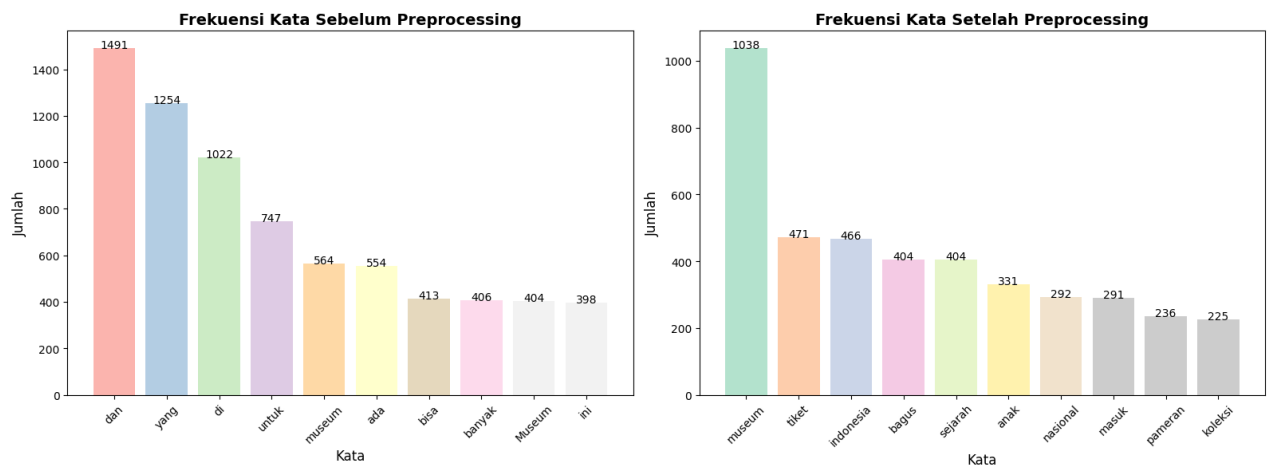
# AFTER
all_text_after = clean_text_column(df["stopword removal"])
words_after = all_text_after.split()
filtered_words_after = [word for word in words_after if word.lower() not in stopwords]
word_counts_after = Counter(filtered_words_after)
top_words_after = word_counts_after.most_common(10)
word_after, count_after = zip(*top_words_after)

# Plot berdampingan
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# BEFORE Plot
colors_before = plt.cm.Pastel1(range(len(word_before)))
bars1 = axes[0].bar(word_before, count_before, color=colors_before)
axes[0].set_title("Frekuensi Kata Sebelum Preprocessing", fontsize=14, fontweight='bold')
axes[0].set_xlabel("Kata", fontsize=12)
axes[0].set_ylabel("Jumlah", fontsize=12)
axes[0].tick_params(axis='x', rotation=45)
for bar, count in zip(bars1, count_before):
    axes[0].text(bar.get_x() + bar.get_width()/2, count + 0.1, str(count), ha='center')

# AFTER Plot
colors_after = plt.cm.Pastel2(range(len(word_after)))
bars2 = axes[1].bar(word_after, count_after, color=colors_after)
axes[1].set_title("Frekuensi Kata Setelah Preprocessing", fontsize=14, fontweight='bold')
axes[1].set_xlabel("Kata", fontsize=12)
axes[1].set_ylabel("Jumlah", fontsize=12)
axes[1].tick_params(axis='x', rotation=45)
for bar, count in zip(bars2, count_after):
    axes[1].text(bar.get_x() + bar.get_width()/2, count + 0.1, str(count), ha='center')

plt.tight_layout()
plt.show()
```



```
df.to_csv('Hasil_Preprocessing_Data.csv', encoding='utf8', index=False);
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1584 entries, 0 to 1616
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   text             1584 non-null   object
1   cleaning         1584 non-null   object
2   case_folding     1584 non-null   object
3   normalisasi     1584 non-null   object
4   tokenize         1584 non-null   object
```

```
5  stopword removal  1584 non-null  object
dtypes: object(6)
memory usage: 86.6+ KB
```

PELEBALAN DATA METODE LAXICON BASED 2 CLASS

```
import pandas as pd
```

```
data = pd.read_csv("Hasil_Preprocessing_Data.csv")
data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1584 entries, 0 to 1583
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0    text                  1584 non-null  object
1    cleaning              1584 non-null  object
2    case_folding          1584 non-null  object
3    normalisasi           1584 non-null  object
4    tokenize              1584 non-null  object
5    stopword removal      1575 non-null  object
dtypes: object(6)
memory usage: 74.4+ KB
```

	text	cleaning	case_folding	normalisasi	tokenize	stopword removal
0	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari HUT TNI jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	kebetulan saya ke monas pas hari hut tni jadi ...	['kebetulan', 'saya', 'ke', 'monas', 'pas', 'h...]	monas pas hut tni padat masuk
1	Banyak arca dan cukup menarik untuk anak2 teru...	Banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	banyak arca dan cukup menarik untuk anak terut...	['banyak', 'arca', 'dan', 'cukup', 'menarik', ...]	arca menarik anak ruangan imersifnya
2	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	sangatt seru sekali liburan bersama keluarga k...	['sangatt', 'seru', 'sekali', 'liburan', 'bers...]	sangatt seru liburan keluarga kesini benda ber...
3	Pengalaman yang menyenangkan	Pengalaman yang menyenangkan	pengalaman yang menyenangkan	pengalaman yang menyenangkan	['pengalaman', 'yang', 'menyenangkan']	pengalaman menyenangkan

```
data = data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1575 entries, 0 to 1583
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  ---
0    text                  1575 non-null  object
1    cleaning              1575 non-null  object
2    case_folding          1575 non-null  object
3    normalisasi           1575 non-null  object
4    tokenize              1575 non-null  object
5    stopword removal      1575 non-null  object
dtypes: object(6)
memory usage: 86.1+ KB
```

```
data = pd.DataFrame(data[['stopword removal']])
data.head(5)
```

	stopword removal
0	monas pas hut tni padat masuk
1	arca menarik anak ruangan imersifnya
2	sangatt seru liburan keluarga kesini benda ber...
3	pengalaman menyenangkan
4	rekomendasi alternatif liburan edukasi

```
import pandas as pd
import requests
```

```
# Unduh kamus leksikon positif dan negatif dari GitHub
positive_url = "https://raw.githubusercontent.com/fajri91/InSet/master/positive.tsv"
negative_url = "https://raw.githubusercontent.com/fajri91/InSet/master/negative.tsv"

positive_lexicon = set(pd.read_csv(positive_url, sep="\t", header=None)[0])
negative_lexicon = set(pd.read_csv(negative_url, sep="\t", header=None)[0])
```

```
def determine_sentiment(text):
    if isinstance(text, str):
        positive_count = sum(1 for word in text.split() if word in positive_lexicon)
        negative_count = sum(1 for word in text.split() if word in negative_lexicon)
        sentiment_score = positive_count - negative_count
        sentiment = "Negatif" if sentiment_score <= -1 else "Positif"
        return sentiment_score, sentiment
    return -1, "Negatif"

data[['Score', 'Sentiment']] = data['stopword removal'].apply(lambda x: pd.Series(determine_sentiment(x)))
data.head(20)
```

	stopword removal	Score	Sentiment
0	monas pas hut tni padat masuk	0	Positif
1	arca menarik anak ruangan imersifnya	-2	Negatif
2	sangatt seru liburan keluarga kesini benda ber...	2	Positif
3	pengalaman menyenangkan	1	Positif
4	rekomendasi alternatif liburan edukasi	1	Positif
5	museum nasional anggaran menyebut kumuh museum...	-4	Negatif
6	museum bagus koleksi dewadewa indonesia kuno b...	1	Positif
7	seru banget	1	Positif
8	seumur hidup datanglah kesini mengenali memaha...	1	Positif
9	bersih pelayanan bagus security ramah sopan te...	2	Positif
10	pengalaman anakanak pertunjukan interaktifnya ...	0	Positif
11	museum nasional indonesia masuk ruang tenang p...	0	Positif
12	kerja bagus	1	Positif
13	museum bagus mengenal sejarah indonesia nusantara	1	Positif
14	terawat informatis fasilitas interaktif bosan ...	2	Positif
15	museum cocok belajar sejarah zaman prasejarah ...	1	Positif
16	dibutuhkan jam artefak dipamerkan	0	Positif
17	temukan sejarah nyata indonesia berkunjung kal...	4	Positif
18	belajar sejarah museum	1	Positif
19	museum bagus pecinta sejarah membeli tiket kar...	2	Positif

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

# Menghitung jumlah sentimen

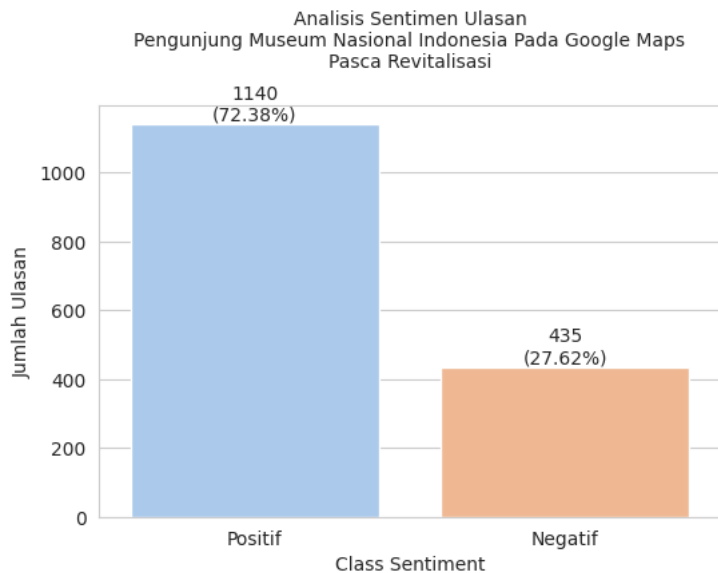
sentiment_count = data['Sentiment'].value_counts()
sns.set_style('whitegrid')

fig, ax = plt.subplots(figsize=(6, 4))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette='pastel')
plt.title('Analisis Sentimen Ulasan\nPengunjung Museum Nasional Indonesia Pada Google Maps\nPasca Revitalisasi', fontsize=12)
plt.xlabel('Class Sentiment', fontsize=10)
plt.ylabel('Jumlah Ulasan', fontsize=10)

total = len(data['Sentiment'])

for i, count in enumerate(sentiment_count.values):
    percentage = f'{100 * count / total:.2f}%'
    ax.text(i, count + 0.10, f'{count}\n({percentage})', ha='center', va='bottom')

plt.show()
```



```
data.to_csv('Hasil_Labelling_Data_Inset_Lax.csv',encoding='utf8', index=False)
```

✓ **Aspect Based Sentiment Classification (ABSC)**

```
import pandas as pd
```

```
data = pd.read_csv("Hasil_Labelling_Data_Inset_Lax.csv")
data.info()
data.head(5)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1575 entries, 0 to 1574
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   stopword removal 1575 non-null   object
1   Score            1575 non-null   int64
2   Sentiment        1575 non-null   object
dtypes: int64(1), object(2)
memory usage: 37.0+ KB
```

	stopword removal	Score	Sentiment
0	monas pas hut tni padat masuk	0	Positif
1	arca menarik anak ruangan imersifnya	-2	Negatif
2	sangatt seru liburan keluarga kesini benda ber...	2	Positif
3	pengalaman menyenangkan	1	Positif
4	rekomendasi alternatif liburan edukasi	1	Positif

```
data = data.dropna()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1575 entries, 0 to 1574
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   stopword removal 1575 non-null   object
1   Score            1575 non-null   int64
2   Sentiment        1575 non-null   object
dtypes: int64(1), object(2)
memory usage: 37.0+ KB
```

```
data = pd.DataFrame(data[['stopword removal', 'Sentiment']])
data.head(5)
```



```

"wisata sejarah", "koleksi sejarah", "koleksi budaya",
"benda bersejarah", "nilai sejarah", "galeri museum",
"pameran tetap", "pameran temporer", "objek bersejarah",
"tata letak", "ruang imersifa", "ruang imersif", "isi sedikit",

# ===== 3 KATA =====
"koleksi sejarah lengkap",
"wisata edukasi sejarah", "benda peninggalan sejarah",
"koleksi budaya indonesia", "pameran sejarah menarik",
"museum sejarah nasional", "koleksi museum lengkap",
]
}

# daftar semua aspek yang ada
all_aspects = list(aspect_keywords.keys())
counter = 0 # untuk round robin

def aspect_extraction(text):
    global counter
    for aspect, keywords in aspect_keywords.items():
        if any(keyword in text for keyword in keywords):
            return aspect
    # jika tidak ketemu keyword -> dibagi rata
    aspect = all_aspects[counter % len(all_aspects)]
    counter += 1
    return aspect

# terapkan ke dataframe
data['aspect_extraction'] = data['stopword removal'].astype(str).apply(aspect_extraction)
data.head(20)

```

	stopword removal	Sentiment	aspect_extraction
0	monas pas hut tni padat masuk	Positif	Aksesibilitas\n(Accesibility)
1	arca menarik anak ruangan imersifnya	Negatif	Fasilitas\n(Amenities)
2	sangatt seru liburan keluarga kesini benda ber...	Positif	Koleksi\n(collection)
3	pengalaman menyenangkan	Positif	Pelayanan\n(Service)
4	rekomendasi alternatif liburan edukasi	Positif	Koleksi\n(collection)
5	museum nasional anggaran menyebut kumuh museum...	Negatif	Koleksi\n(collection)
6	museum bagus koleksi dewadewa indonesia kuno b...	Positif	Koleksi\n(collection)
7	seru banget	Positif	Fasilitas\n(Amenities)
8	seumur hidup datanglah kesini mengenali memaha...	Positif	Pelayanan\n(Service)
9	bersih pelayanan bagus security ramah sopan te...	Positif	Fasilitas\n(Amenities)
10	pengalaman anakanak pertunjukan interaktifnya ...	Positif	Pelayanan\n(Service)
11	museum nasional indonesia masuk ruang tenang p...	Positif	Aksesibilitas\n(Accesibility)
12	kerja bagus	Positif	Koleksi\n(collection)
13	museum bagus mengenal sejarah indonesia nusantara	Positif	Koleksi\n(collection)
14	terawat informatis fasilitas interaktif bosan ...	Positif	Pelayanan\n(Service)
15	museum cocok belajar sejarah zaman prasejarah ...	Positif	Koleksi\n(collection)
16	dibutuhkan jam artefak dipamerkan	Positif	Koleksi\n(collection)
17	temukan sejarah nyata indonesia berkunjung kal...	Positif	Koleksi\n(collection)
18	belajar sejarah museum	Positif	Koleksi\n(collection)
19	museum bagus pecinta sejarah membeli tiket kar...	Positif	Fasilitas\n(Amenities)

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sentiment_count = data['aspect_extraction'].value_counts()
sns.set_style('whitegrid')

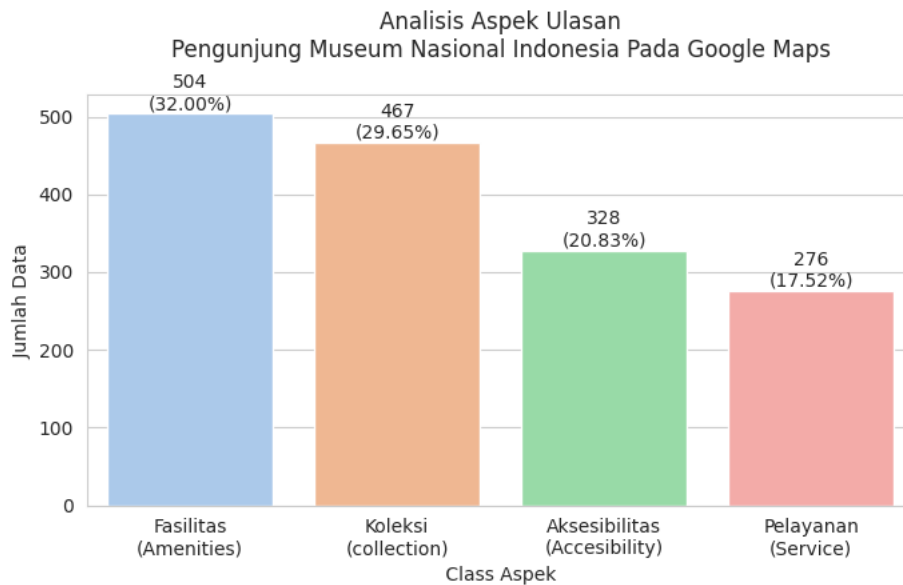
fig, ax = plt.subplots(figsize=(8, 4))
ax = sns.barplot(x=sentiment_count.index, y=sentiment_count.values, palette='pastel')
plt.title('Analisis Aspek Ulasan\nPengunjung Museum Nasional Indonesia Pada Google Maps', fontsize=12, pad=20)
plt.xlabel(' Class Aspek ', fontsize=10)
plt.ylabel(' Jumlah Data ', fontsize=10)

```

```
total = len(data['aspect_extraction'])

for i, count in enumerate(sentiment_count.values):
    percentage = f'{100 * count / total:.2f}%'
    ax.text(i, count + 0.10, f'{count}\n({percentage})', ha='center', va='bottom')

plt.show()
```



```
data.to_csv('Hasil_AspectExtraction_Data.csv', encoding='utf8', index=False)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1575 entries, 0 to 1574
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   stopword removal 1575 non-null   object
1   Sentiment        1575 non-null   object
2   aspect_extraction 1575 non-null   object
dtypes: object(3)
memory usage: 37.0+ KB
```

Start coding or [generate](#) with AI.

✓ **WORDCLOUD & FREKUENSI KATA NGRAM**

```
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import CountVectorizer
from wordcloud import WordCloud

# Ganti dengan path dataset kamu
df = pd.read_csv("Hasil_AspectExtraction_Data.csv")

df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1575 entries, 0 to 1574
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   stopword removal      1575 non-null   object
1   Sentiment              1575 non-null   object
2   aspect_extraction      1575 non-null   object
dtypes: object(3)
memory usage: 37.0+ KB
```

	stopword removal	Sentiment	aspect_extraction
0	monas pas hut tni padat masuk	Positif	Aksesibilitas\n(Accesibility)
1	arca menarik anak ruangan imersifnya	Negatif	Fasilitas\n(Amenities)
2	sangatt seru liburan keluarga kesini benda ber...	Positif	Koleksi\n(collection)
3	pengalaman menyenangkan	Positif	Pelayanan\n(Service)
4	rekomendasi alternatif liburan edukasi	Positif	Koleksi\n(collection)

```
# Pastikan tidak ada nilai kosong
df = df.dropna(subset=['stopword removal', 'Sentiment'])
df.head()
```

	stopword removal	Sentiment	aspect_extraction
0	monas pas hut tni padat masuk	Positif	Aksesibilitas\n(Accesibility)
1	arca menarik anak ruangan imersifnya	Negatif	Fasilitas\n(Amenities)
2	sangatt seru liburan keluarga kesini benda ber...	Positif	Koleksi\n(collection)
3	pengalaman menyenangkan	Positif	Pelayanan\n(Service)
4	rekomendasi alternatif liburan edukasi	Positif	Koleksi\n(collection)

```
def wordcloud_ngram(data, sentiment_label, ngram_range=(2,3), max_words=1000):
    text_data = data[data['Sentiment'] == sentiment_label]['stopword removal']
    text_data = text_data.dropna()
    text_data = text_data[text_data.str.len() > 0]

    vectorizer = CountVectorizer(ngram_range=ngram_range)
    X = vectorizer.fit_transform(text_data)

    word_freq = dict(zip(
        vectorizer.get_feature_names_out(),
        X.sum(axis=0).A1
    ))

    # filter frekuensi kecil
    word_freq = {k: v for k, v in word_freq.items() if v > 2}

    wc = WordCloud(
        width=1000,
        height=500,
        background_color='white',
        max_words=max_words,
        prefer_horizontal=0.9,
        relative_scaling=0.5,
        collocations=False,
        min_font_size=8,
        max_font_size=120,
        scale=5
    ).generate_from_frequencies(word_freq)

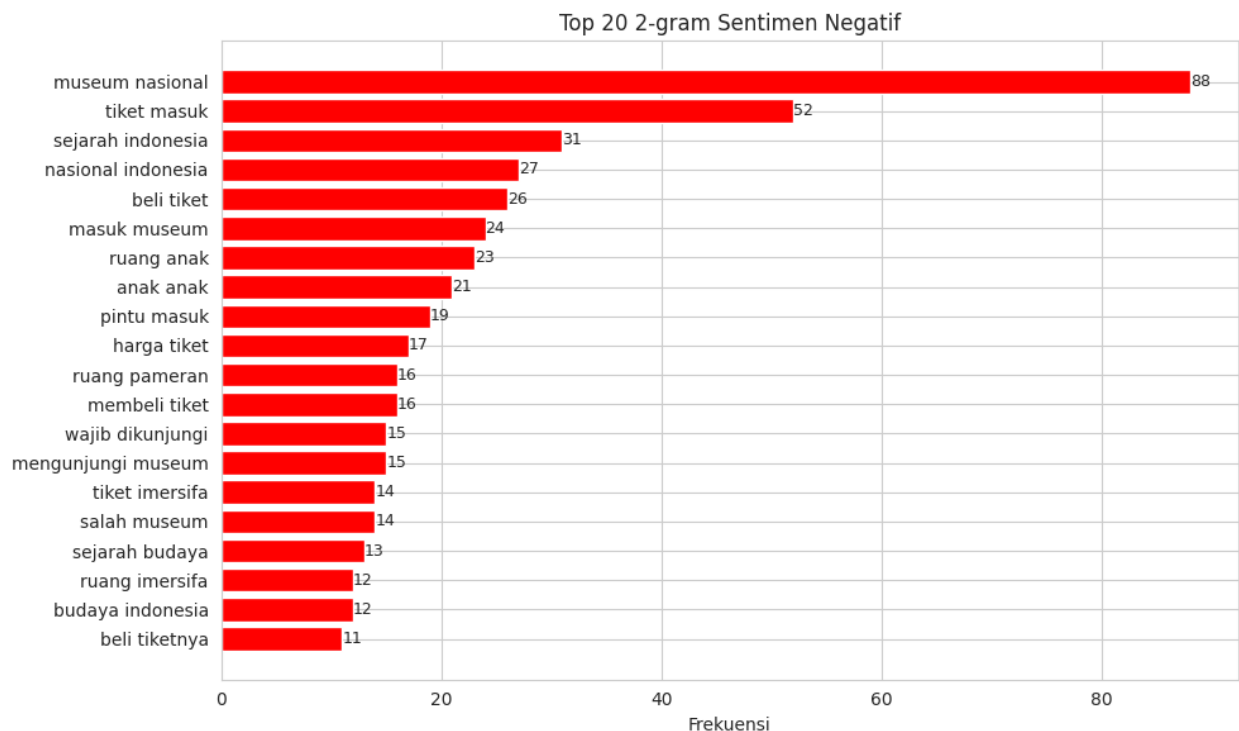
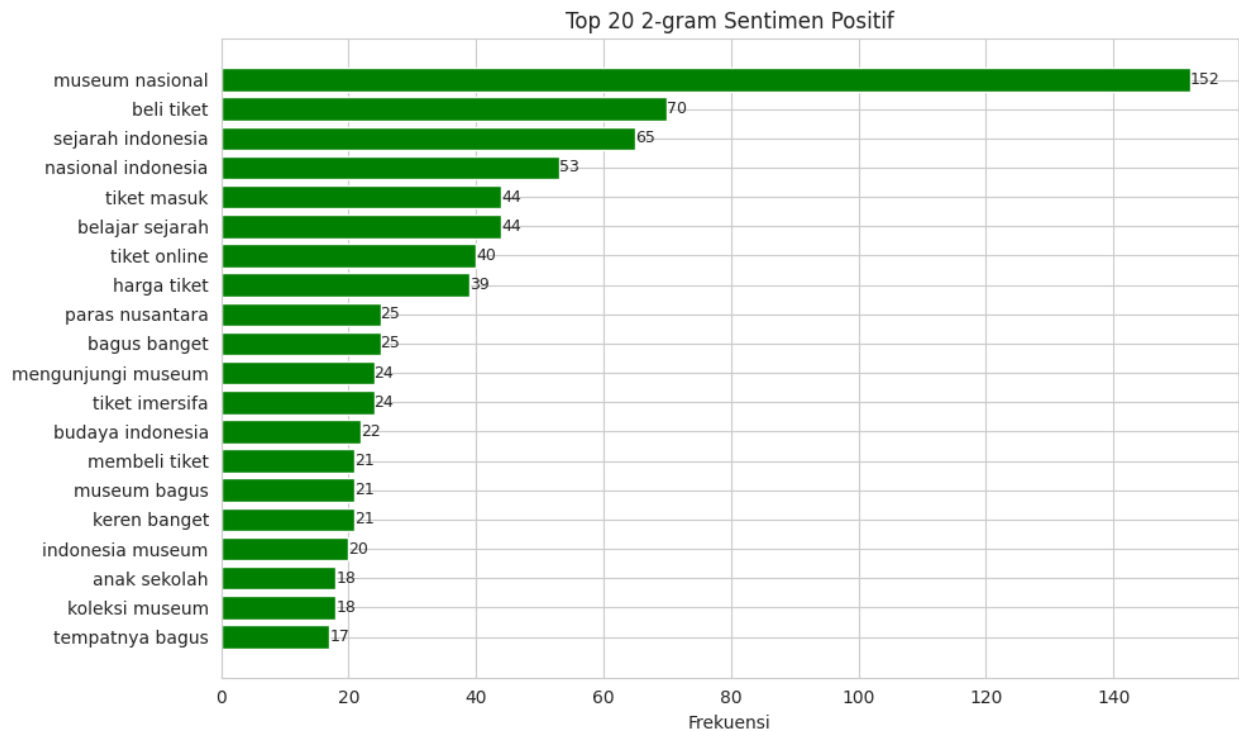
    plt.figure(figsize=(8,4))
    plt.imshow(wc, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'WordCloud {ngram_range[0]}-gram Sentimen {sentiment_label}', fontsize=14)
    plt.show()

# Bigram
wordcloud_ngram(df, 'Positif', ngram_range=(2,2))
print()
wordcloud_ngram(df, 'Negatif', ngram_range=(2,2))
```



17/21

```
# Pemanggilan
plot_top_ngram(df, 'Positif', ngram_range=(2,2), color='green')
print()
plot_top_ngram(df, 'Negatif', ngram_range=(2,2), color='red')
```



✓ **PENERAPAN ALGORITMA SVM - Kernal Linear**

```
import pandas as pd

file_path = 'Hasil_AspectExtraction_Data.csv'
data = pd.read_csv(file_path)

data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1575 entries, 0 to 1574
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   stopword removal      1575 non-null   object
1   Sentiment              1575 non-null   object
2   aspect_extraction      1575 non-null   object
dtypes: object(3)
memory usage: 37.0+ KB
```

	stopword removal	Sentiment	aspect_extraction
0	monas pas hut tni padat masuk	Positif	Aksesibilitas\n(Accesibility)
1	arca menarik anak ruangan imersifnya	Negatif	Fasilitas\n(Amenities)
2	sangatt seru liburan keluarga kesini benda ber...	Positif	Koleksi\n(collection)
3	pengalaman menyenangkan	Positif	Pelayanan\n(Service)
4	rekomendasi alternatif liburan edukasi	Positif	Koleksi\n(collection)

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer

cleaned_data = data.dropna(subset=['stopword removal'])

X = cleaned_data['stopword removal']
y = cleaned_data['Sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

print("Jumlah data latih:", len(X_train))
print("Jumlah data uji:", len(X_test))
print('=====')
```

```
Jumlah data latih: 1260
Jumlah data uji: 315
=====
```

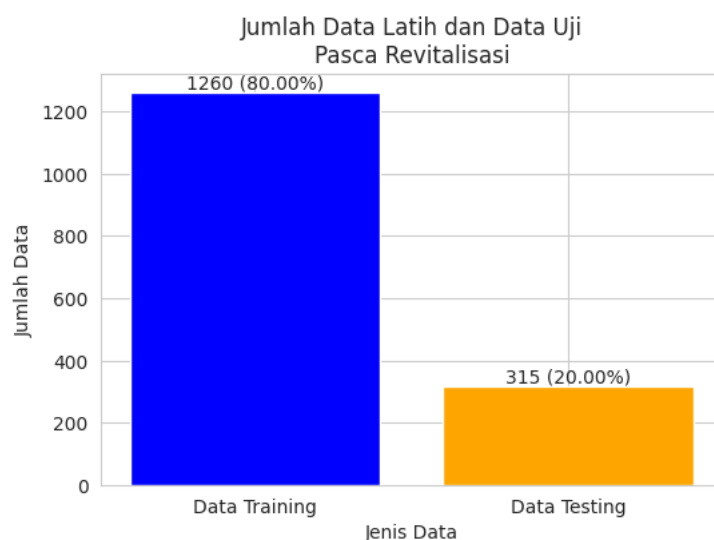
```
import matplotlib.pyplot as plt

train_size = len(X_train)
test_size = len(X_test)

plt.figure(figsize=(6, 4))
bars = plt.bar(['Data Training', 'Data Testing'], [train_size, test_size], color=['blue', 'orange'])

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height} ({height / (train_size + test_size) * 100:.2f}%)',
             ha='center', va='bottom')

plt.title('Jumlah Data Latih dan Data Uji\nPasca Revitalisasi')
plt.xlabel('Jenis Data')
plt.ylabel('Jumlah Data')
plt.show()
```



```

from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

# Inisialisasi TF-IDF Vectorizer
vectorizer = TfidfVectorizer()

# Fit & transform data train
X_train_vec = vectorizer.fit_transform(X_train)

# Transform data test (tanpa fit ulang)
X_test_vec = vectorizer.transform(X_test)

# Cek shape
print("Shape of X_train_vec:", X_train_vec.shape)
print("Shape of X_test_vec:", X_test_vec.shape)

# Lihat hasil vektorisasi (array)
print("\nX_train_vec (as array):")
print(X_train_vec.toarray())

print("\nX_test_vec (as array):")
print(X_test_vec.toarray())

# Konversi ke DataFrame
df_X_train_vec = pd.DataFrame(
    X_train_vec.toarray(),
    columns=vectorizer.get_feature_names_out()
)

# Simpan ke CSV
df_X_train_vec.to_csv('tfidf_vector.csv', encoding='utf-8', index=False)

```

```

Shape of X_train_vec: (1260, 4374)
Shape of X_test_vec: (315, 4374)

```

```

X_train_vec (as array):
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

```

```

X_test_vec (as array):
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

```

```

from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.neural_network import MLPClassifier
import seaborn as sns
import matplotlib.pyplot as plt

# Initialize models
models = {
    "SVM": SVC(kernel='linear', random_state=42),
}

# Train models
results = {}
for model_name, model in models.items():
    model.fit(X_train_vec, y_train)
    y_pred = model.predict(X_test_vec)
    results[model_name] = {
        "accuracy": accuracy_score(y_test, y_pred),
        "classification_report": classification_report(y_test, y_pred, output_dict=True),
        "confusion_matrix": confusion_matrix(y_test, y_pred)
    }

```

```

import matplotlib.pyplot as plt
import seaborn as sns

for model_name, result in results.items():
    fig, ax = plt.subplots(figsize=(6, 4))
    sns.heatmap(
        result["confusion_matrix"],

```

```
annot=True,
fmt='d',
cmap="YlGnBu",
cbar=False,
xticklabels=['Negatif','Positif'],
yticklabels=['Negatif','Positif'],
ax=ax,
square=True,
linewidths=0.5
)
print(f"\nConfusion Matrix for {model_name}:")
ax.set_title(f"{model_name} Confusion Matrix", fontsize=12)
ax.set_xlabel("Predicted", fontsize=12)
ax.set_ylabel("Actual", fontsize=12)

plt.tight_layout()
plt.show()
```

Confusion Matrix for SVM:

