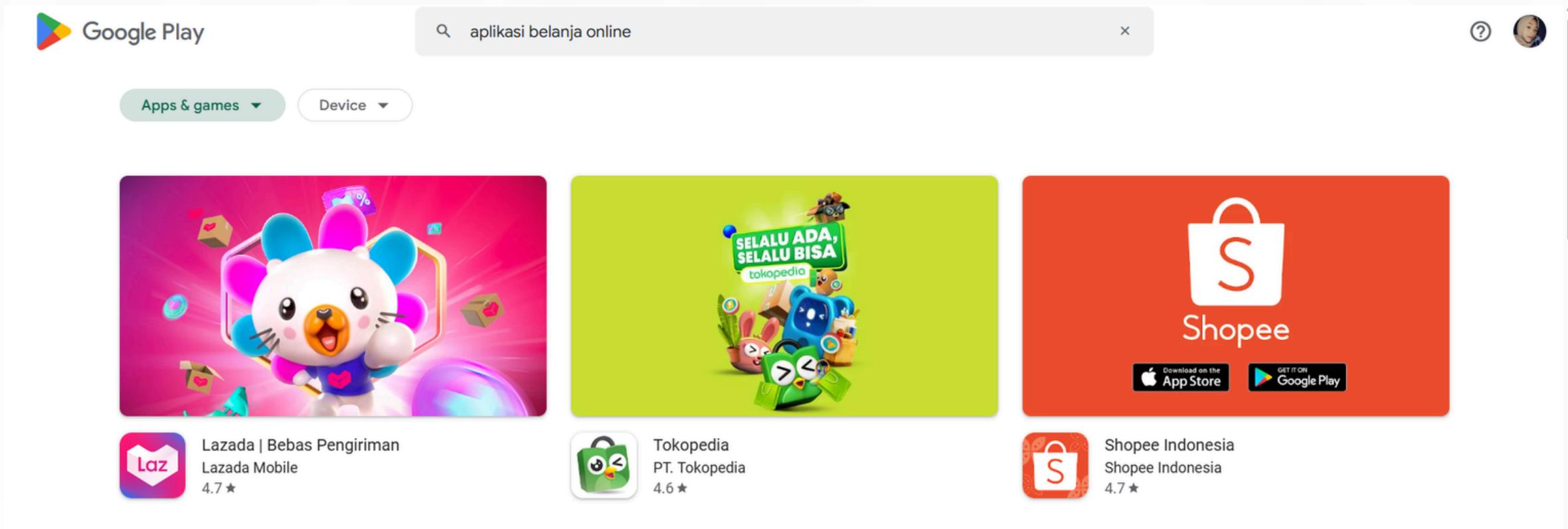


Analisis Sentimen Ulasan Pengguna pada Aplikasi Belanja Online (Shopee, Tokopedia, dan Lazada)

Siti Inayatul Mufidah (5220411319)



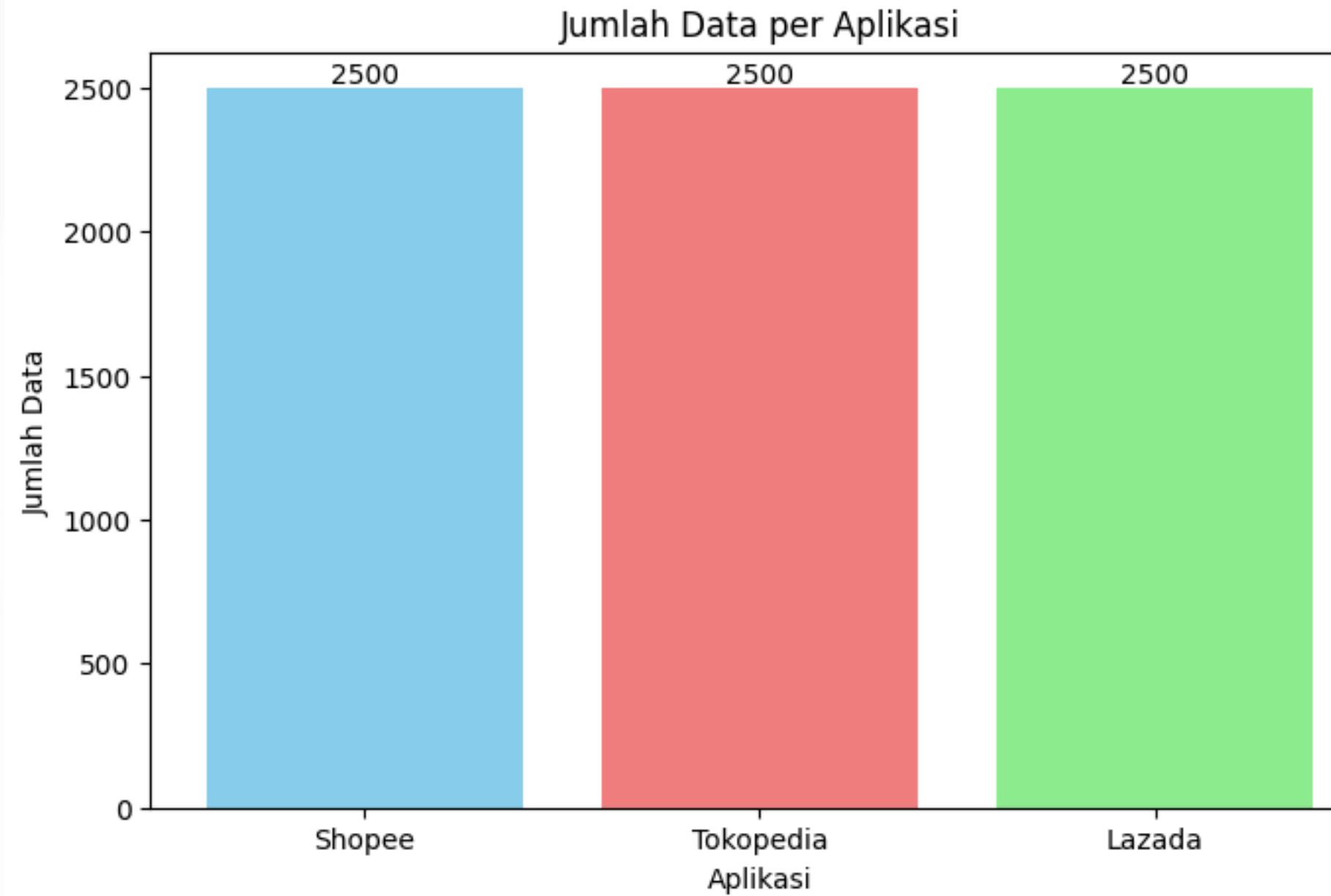
APLIKASI BELANJA ONLINE TERATAS DI PLAYSTORE



Lazada, Tokopedia, dan Shopee merupakan aplikasi belanja online teratas pada playstore.



SCRAPING DATA



Dataset yang digunakan berasal dari hasil scraping ulasan pengguna aplikasi belanja online dengan total 7.500 data ulasan dengan masing-masing aplikasi memiliki jumlah data sebanyak 2.500 data. Data mencakup teks ulasan, skor penilaian, dan nama aplikasi.



PREPROCESSING

Tahapan-tahapan dalam preprocessing dilakukan untuk meningkatkan kualitas data teks sebelum dianalisis. Tahapan preprocessing meliputi:



PREPROCESSING (EKSPLORASI DATA)

Pada proses ini dilakukan pengecekan missing value dan duplikat, dengan tujuan agar data bersih dari data yang berisi nilai kosong dan data yang memiliki duplikat.

```
▶ df.isnull().sum()
```

```
...
```

```
    review 0
```

```
    score 0
```

```
    aplikasi 0
```

```
dtype: int64
```

```
df.duplicated().sum()
```

```
np.int64(2357)
```

Data tidak ada missing value, tetapi data memiliki data yang duplikat sebanyak 2357. Untuk menangani data yang duplikat tersebut dilakukan penghapusan data.

```
df = df.drop_duplicates()
```



PREPROCESSING (CASE FOLDING AND TOKENIZING)

Case folding merupakan salah satu bentuk text preprocessing yang bertujuan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil.

```
#case folding  
df['review'] = df['review'].str.lower()
```

Tokenizing/tokenisasi merupakan proses memecah kalimat menjadi perkata.

```
#tokenizing  
df['tokens'] = df['review'].apply(lambda x: x.split())
```

PREPROCESSING (CLEANING)

Pada saat cleaning dilakukan hapus emoji, hapus non-ASCII, hapus angka, hapus tanda baca/punctuation, hapus spasi berlebihan, dan hapus kata kurang dari 3 huruf.

```
!pip install emoji
import emoji
import re
import string
def clean_text(text):
    text = str(text)
    # 1. HAPUS EMOJI
    text = emoji.replace_emoji(text, replace=' ')
    # 2. HAPUS NON-ASCII
    text = re.sub(r'[^\\x00-\\x7F]+', ' ', text)
    # 3. HAPUS ANGKA
    text = re.sub(r'\\d+', ' ', text)
    # 4. HAPUS TANDA BACA / PUNCTUATION
    text = text.translate(str.maketrans(' ', ' ', string.punctuation))
    # 5. HAPUS SPASI BERLEBIH
    text = re.sub(r'\\s+', ' ', text).strip()
return text
```

```
#hapus kata kurang dari 3
df['tokens'] = df['tokens'].apply(
    lambda x: [word for word in x if len(word) > 3]
)
```



PREPROCESSING (NORMALISASI)

Pada proses ini dilakukan normalisasi huruf berulang serta normalisasi kata gaul dan typo

```
#normalisasi kata gaul dan typo
normalisasi_dict = {
    'gk': 'tidak', 'tampa': 'tanpa',
    'ga': 'tidak', 'tak': 'tidak',
    'nggak': 'tidak', 'prnh': 'pernah',
    'tdk': 'tidak', 'dlm': 'dalam',
    'bgt': 'banget', 'dgn': 'dengan',
    'cs': 'customer service', 'tpi': 'tapi',
    'apk': 'aplikasi', 'jd': 'jadi',
    'udh': 'sudah', 'bagu': 'bagus',
    'blm': 'belum', 'jela': 'jelas',
    'sy': 'saya', 'mengagu': 'mengganggu',
    'apan': 'apaan', 'muas': 'puas',
    'klau': 'kalau',
    'gak': 'tidak',
    'trs': 'terus',
    'kalo': 'kalau',
    'spy': 'supaya',
    'skrng': 'sekarang',
    'tbtb': 'tiba-tiba'

}

def normalize_text(text):
    words = text.split()
    normalized_words = [normalisasi_dict.get(word, word) for word in words]
    return ' '.join(normalized_words)

df['review'] = df['review'].apply(normalize_text)

#normalisasi huruf berulang
def remove_double_char(text):
    return re.sub(r'(.+)\1+', r'\1', text)

df['review'] = df['review'].apply(remove_double_char)
```



PREPROCESSING (STOPWORD AND STEMMING)

Stopword merupakan penghapusan kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna

```
#stopword
!pip install Sastrawi
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
stopword_factory = StopWordRemoverFactory()
stopwords = set(stopword_factory.get_stop_words())
# stopword tambahan
domain_stopwords = {
    'aplikasi', 'jadi', 'kalau', 'sudah', 'sering', 'semua',
    'sama', 'buat', 'sini', 'padahal',
    'shope', 'shopee', 'dong',
    'lazada', 'kirim',
    'tokopedia', 'beli', 'pakai', 'belanja', 'barang'
}
all_stopwords = stopwords.union(domain_stopwords)
df['tokens'] = df['tokens'].apply(
    lambda x: [word for word in x if word not in all_stopwords])
```

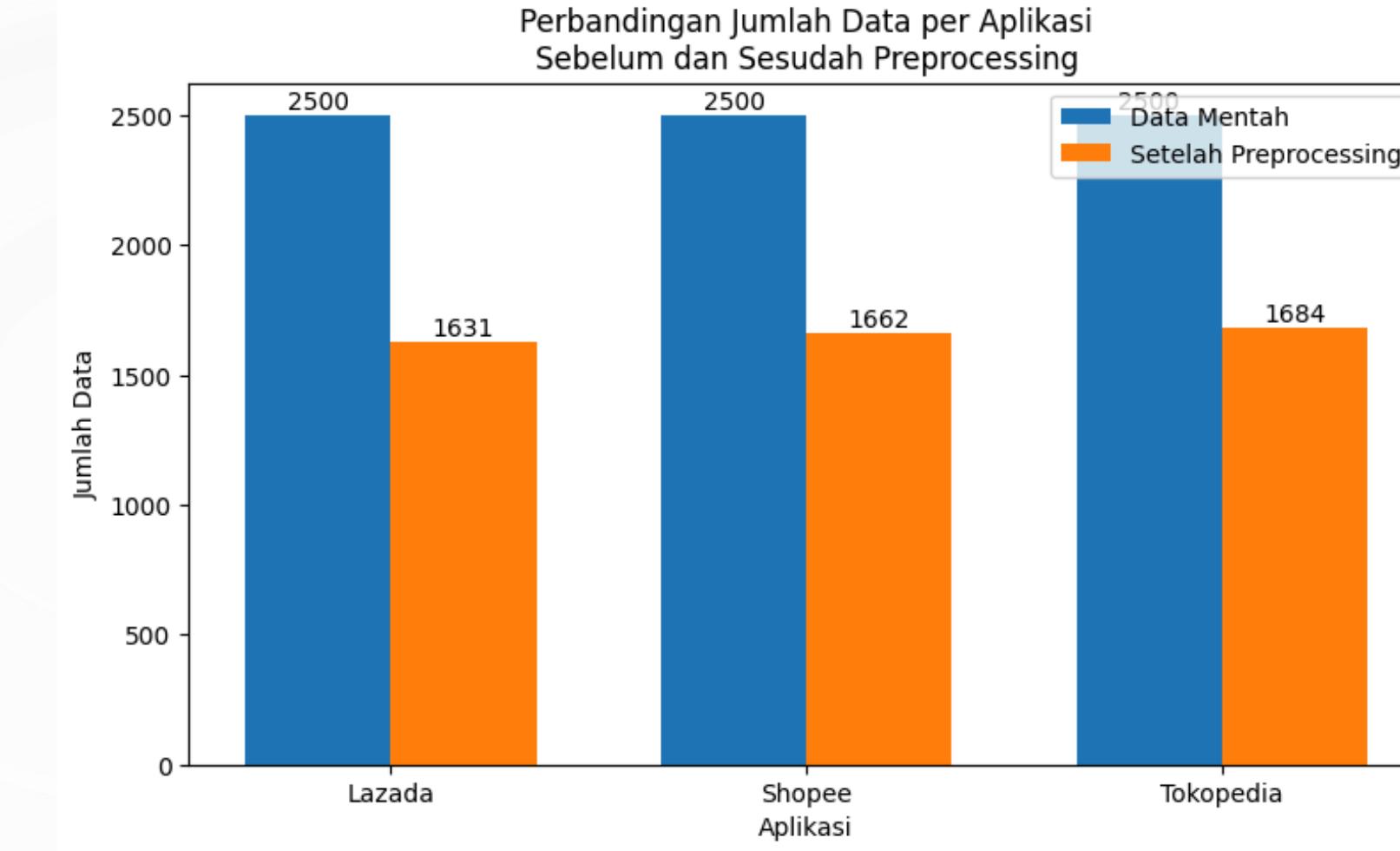
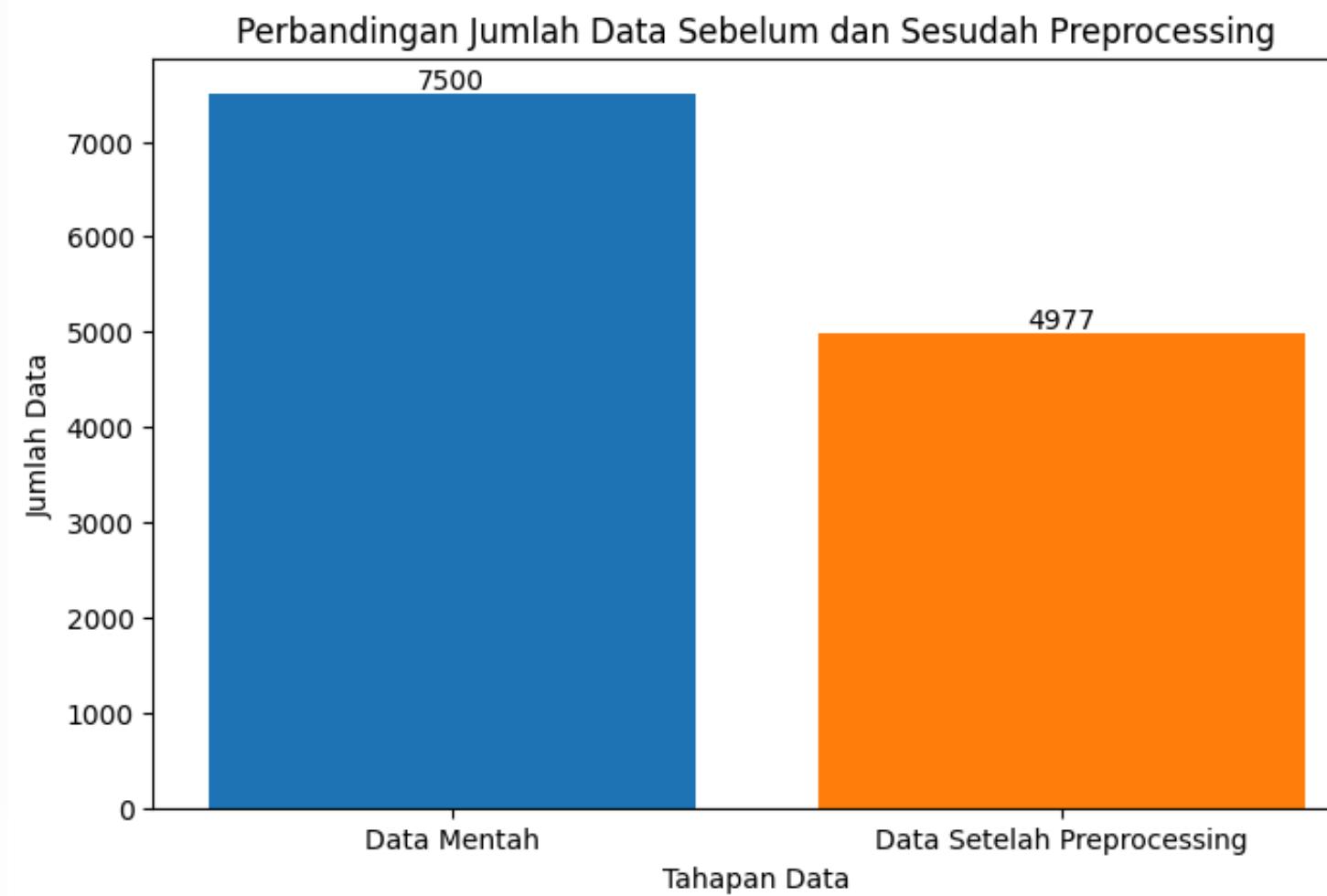
Stemming merupakan proses untuk mengubah kata berimbuhan menjadi kata dasar.

```
#stemming
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
stemmer = StemmerFactory().create_stemmer()
df['tokens'] = df['tokens'].apply(
    lambda x: [stemmer.stem(word) for word in x]
)
```



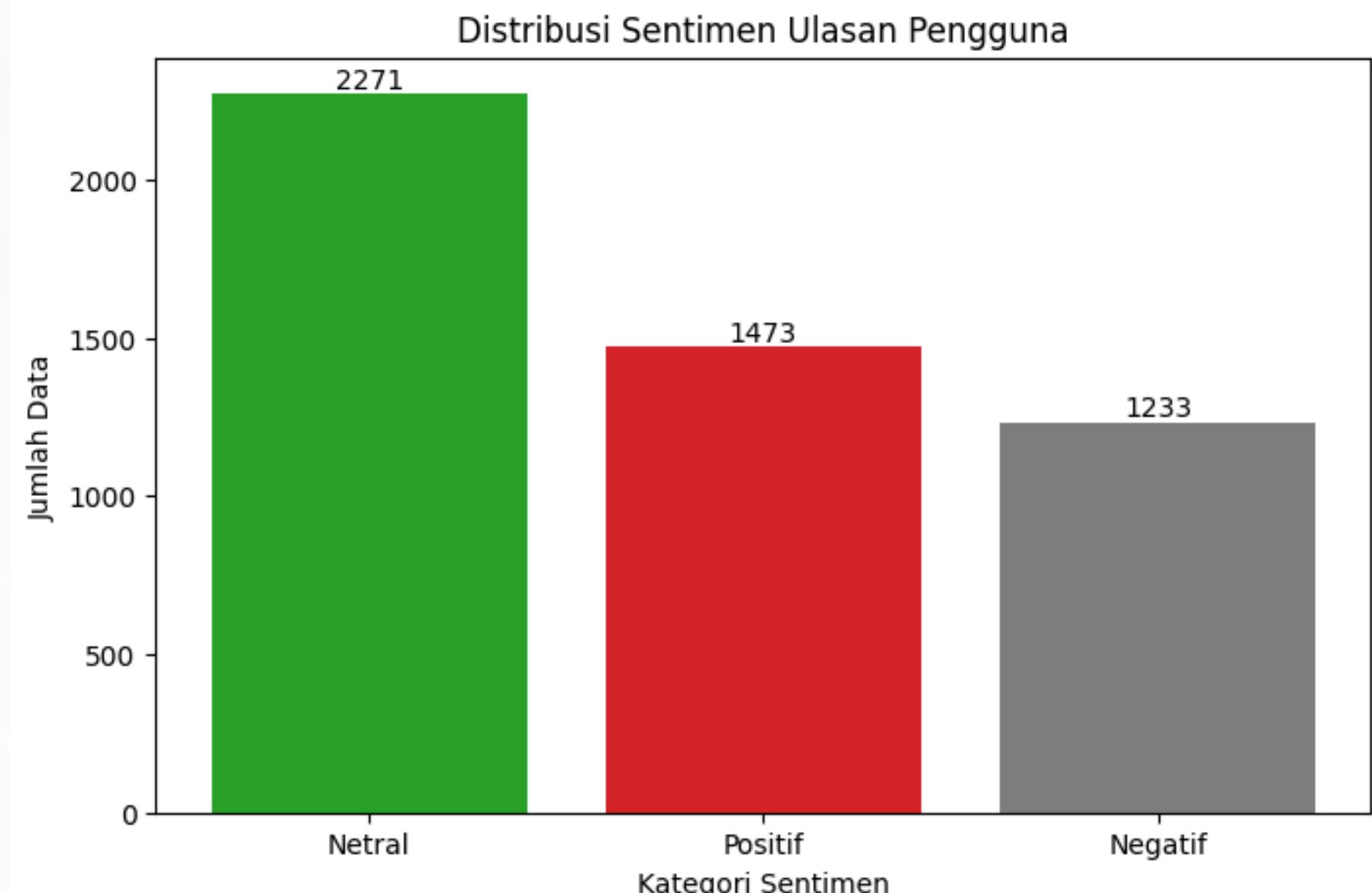
PERBANDINGAN DATA SEBELUM DAN SESUDAH PREPROCESSING

Setelah dilakukan preprocessing data mengalami penurunan.

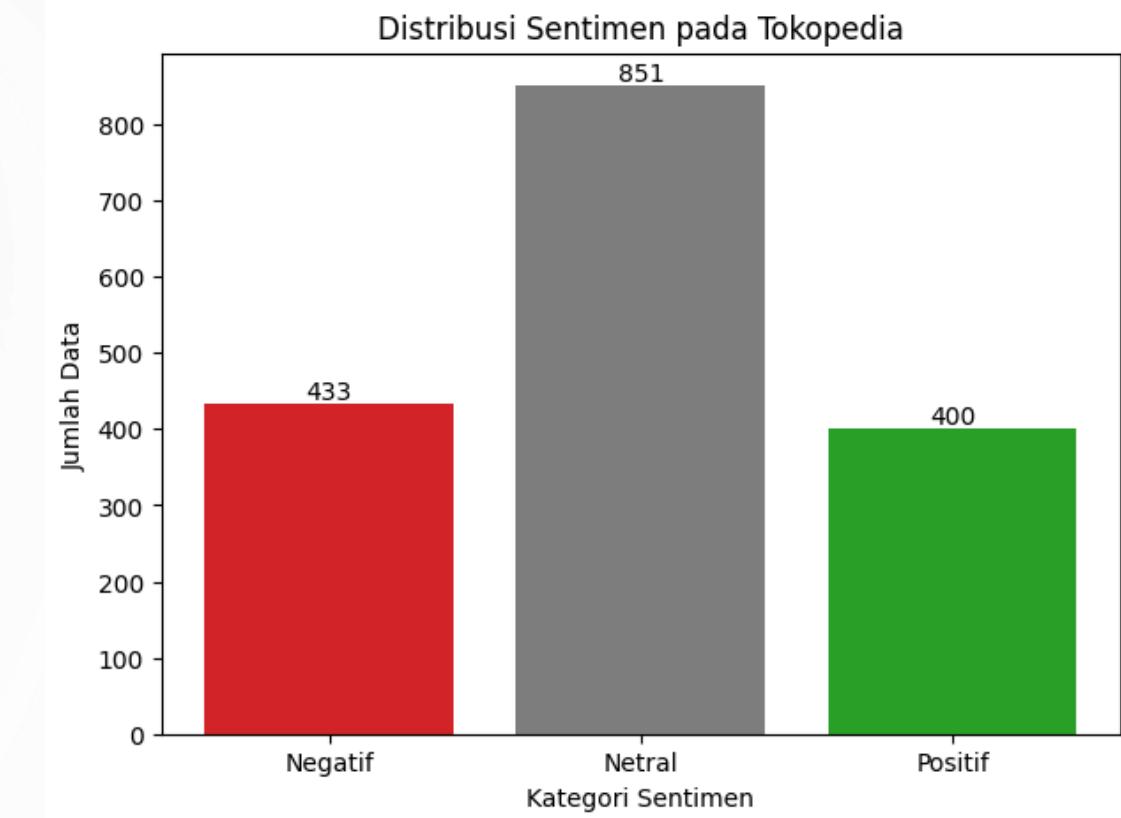
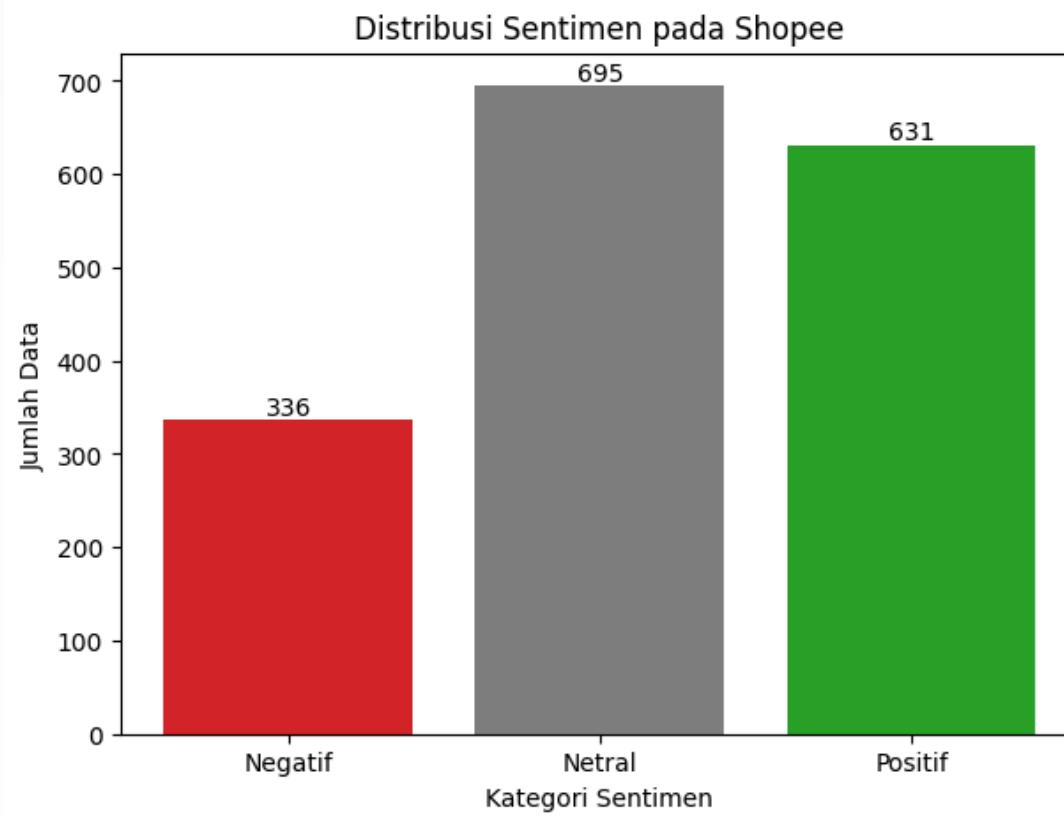
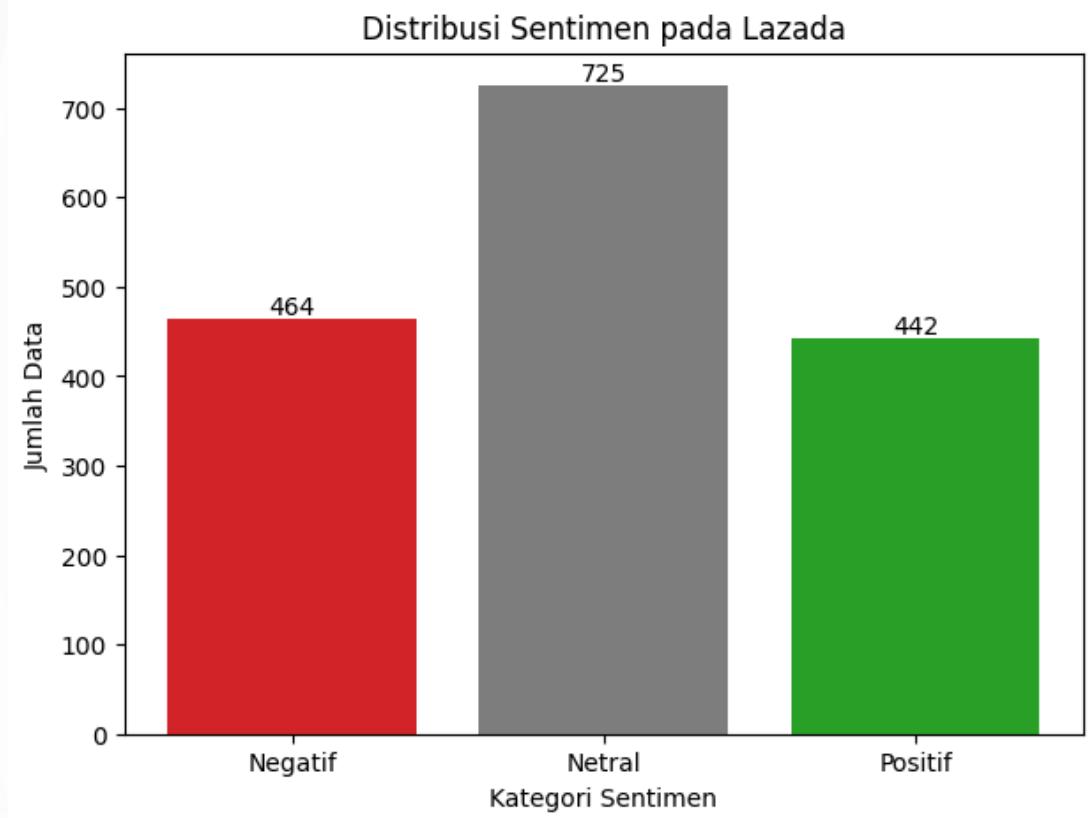


PELABELAN

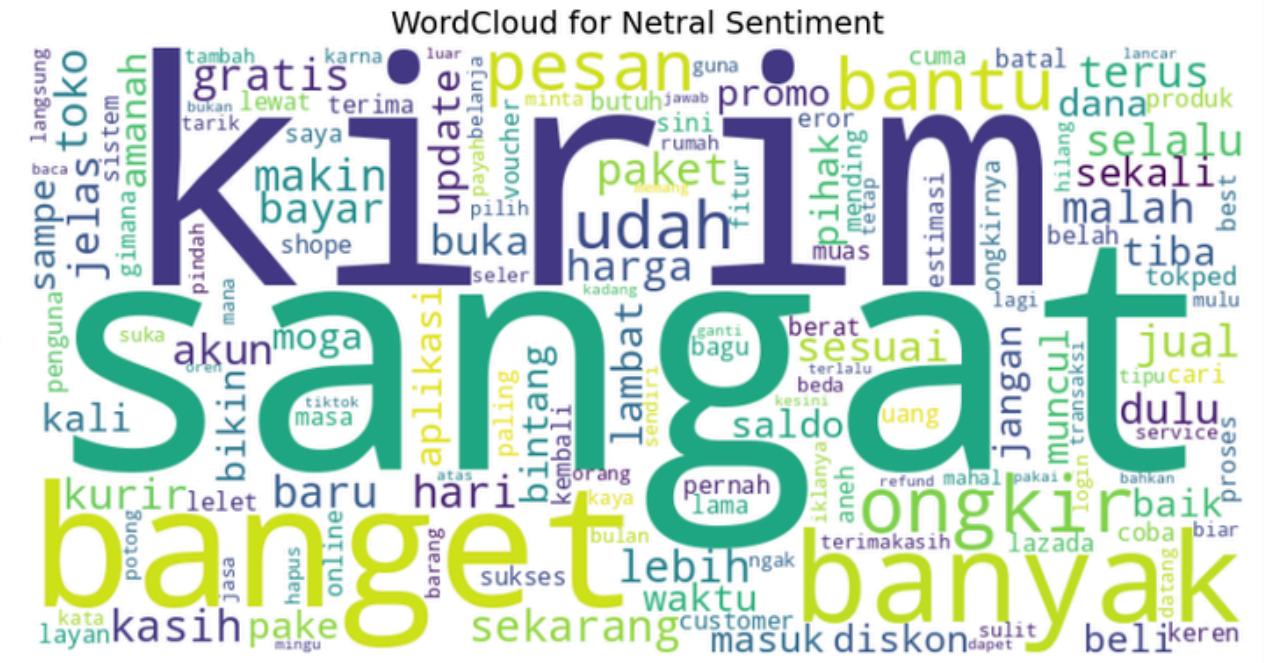
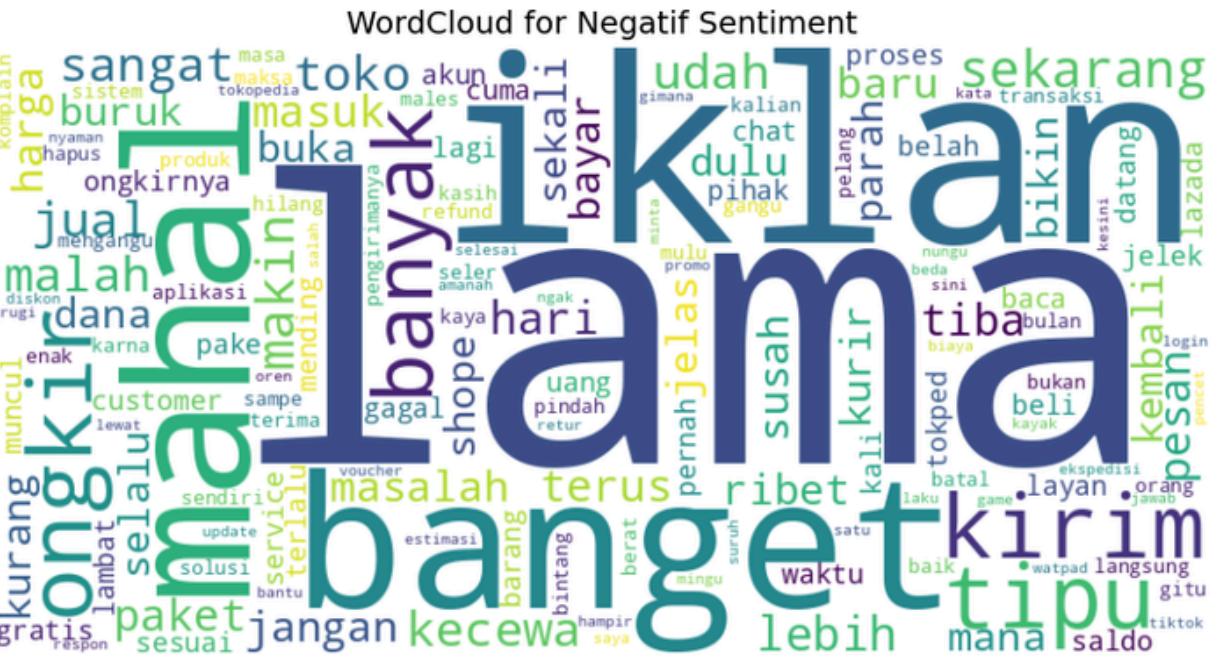
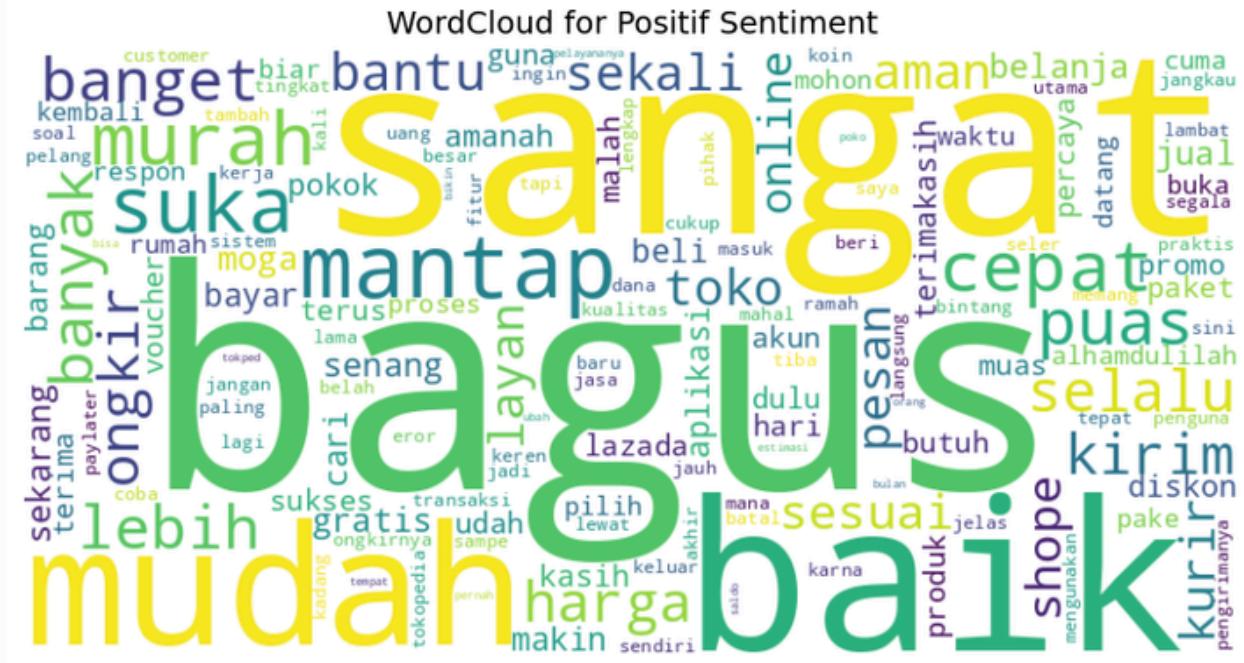
Pelabelan sentimen dilakukan menggunakan metode lexicon-based, yaitu dengan mencocokkan kata-kata pada ulasan pengguna dengan kamus sentimen yang telah ditentukan. Setiap ulasan kemudian diklasifikasikan ke dalam tiga kategori sentimen, yaitu positif, negatif, dan netral, berdasarkan dominasi kata sentimen yang muncul pada teks ulasan.



DISTRIBUSI SENTIMEN



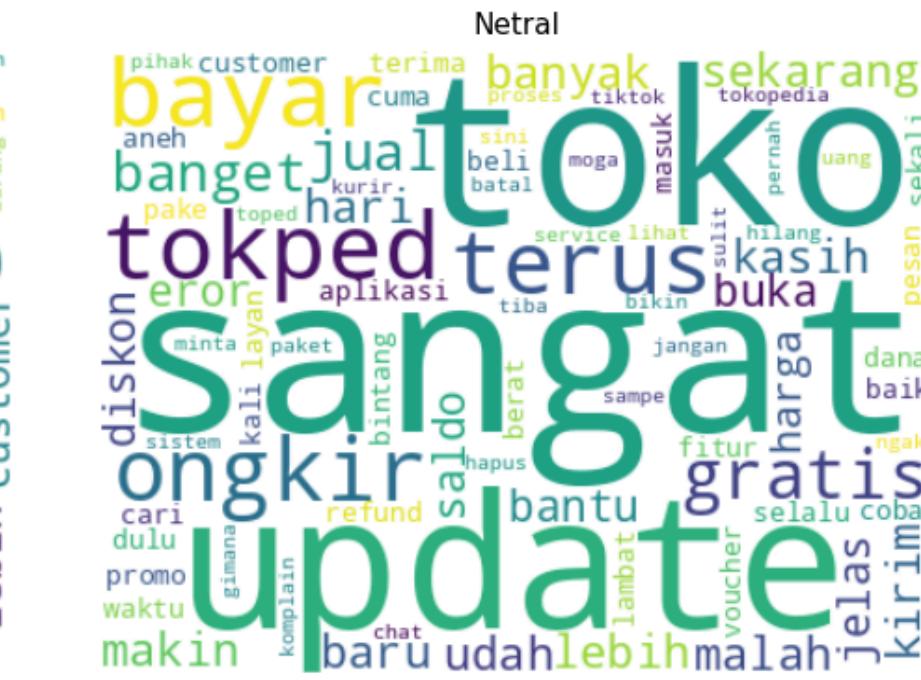
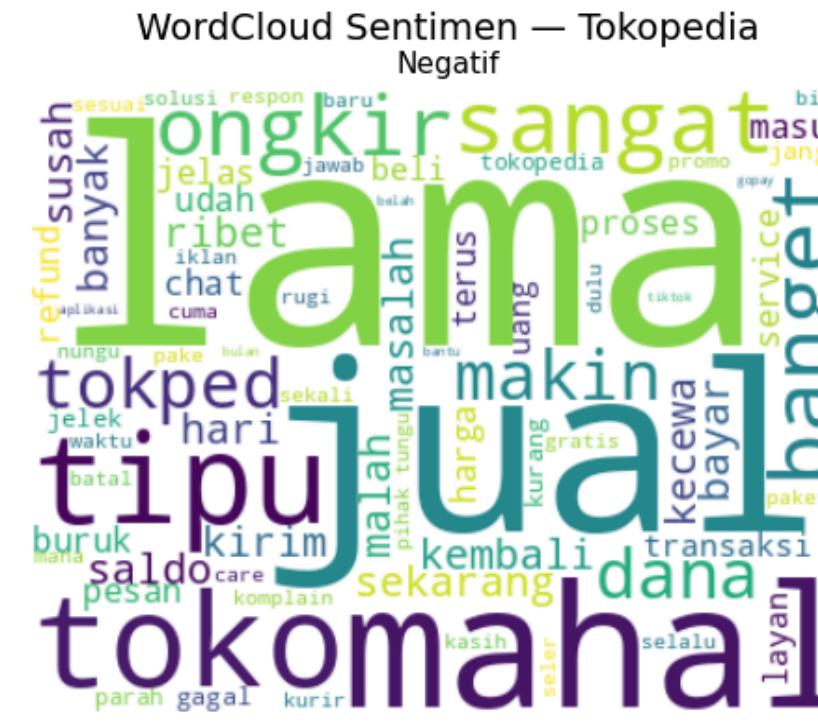
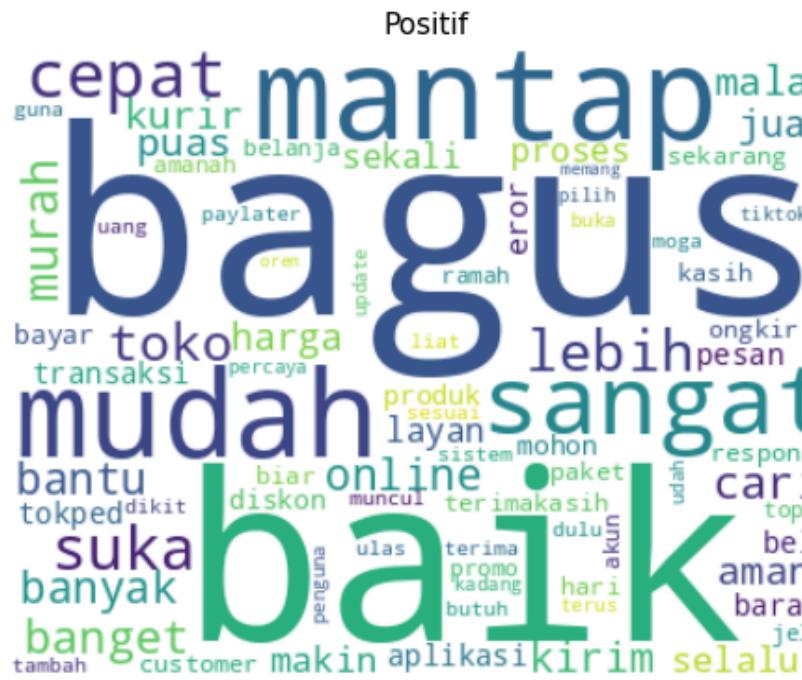
WORDCLOUD SENTIMENT



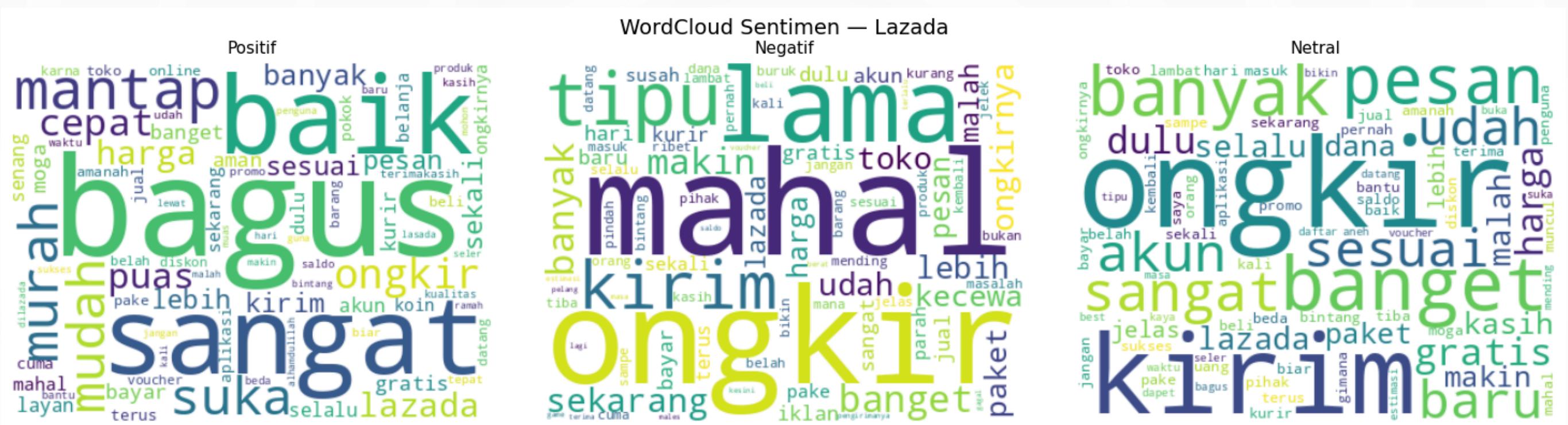
WORDCLOUD SENTIMEN SETIAP APLIKASI (SHOPEE)



WORDCLOUD SENTIMEN SETIAP APLIKASI (TOKOPEDIA)



WORDCLOUD SENTIMEN SETIAP APLIKASI (LAZADA)



Positif

mantap baik bagus sangat suka selalu lazada murah harga amanah mudah puas sangat

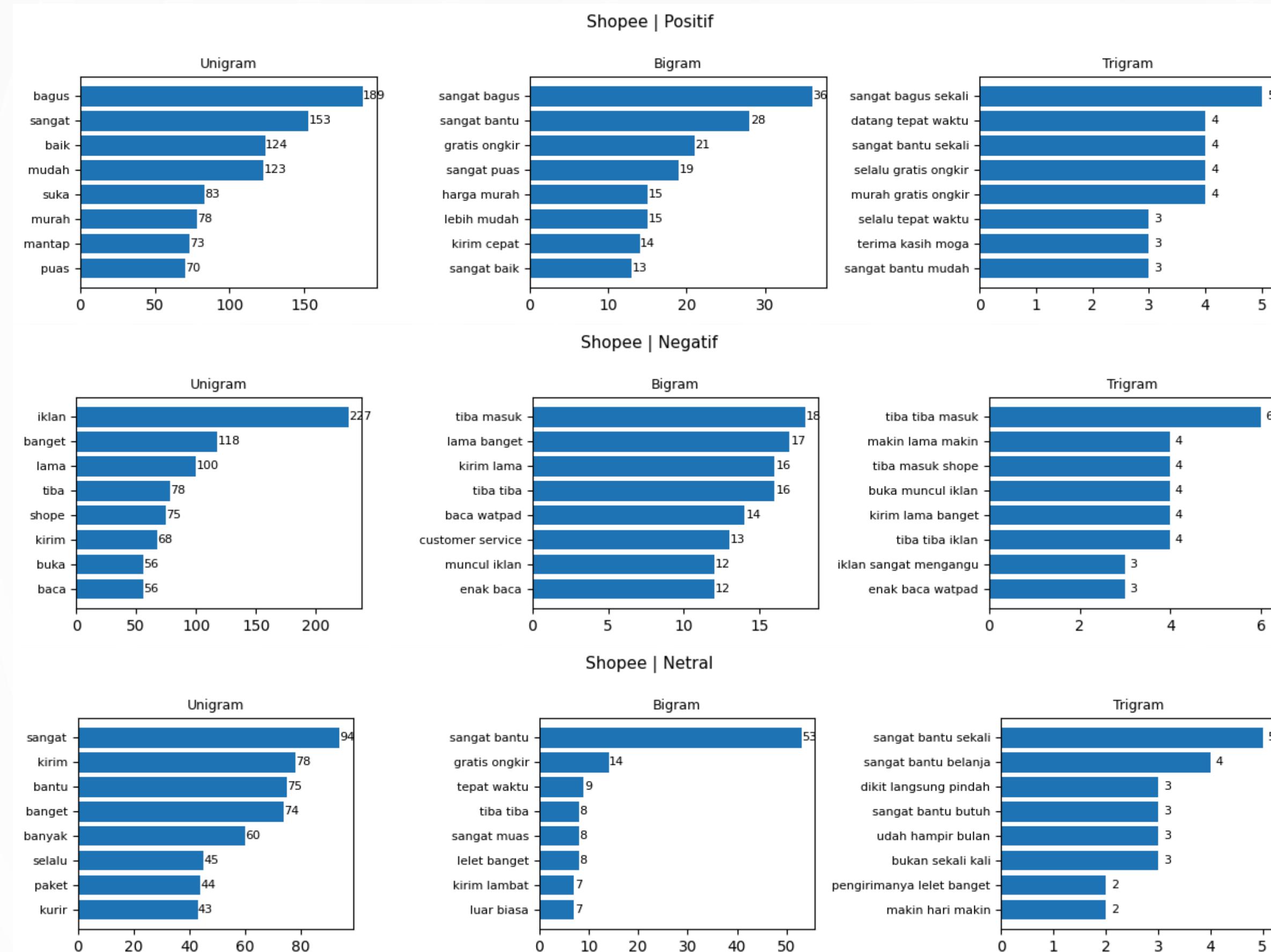
Negatif

tipu lama mahal ongkir

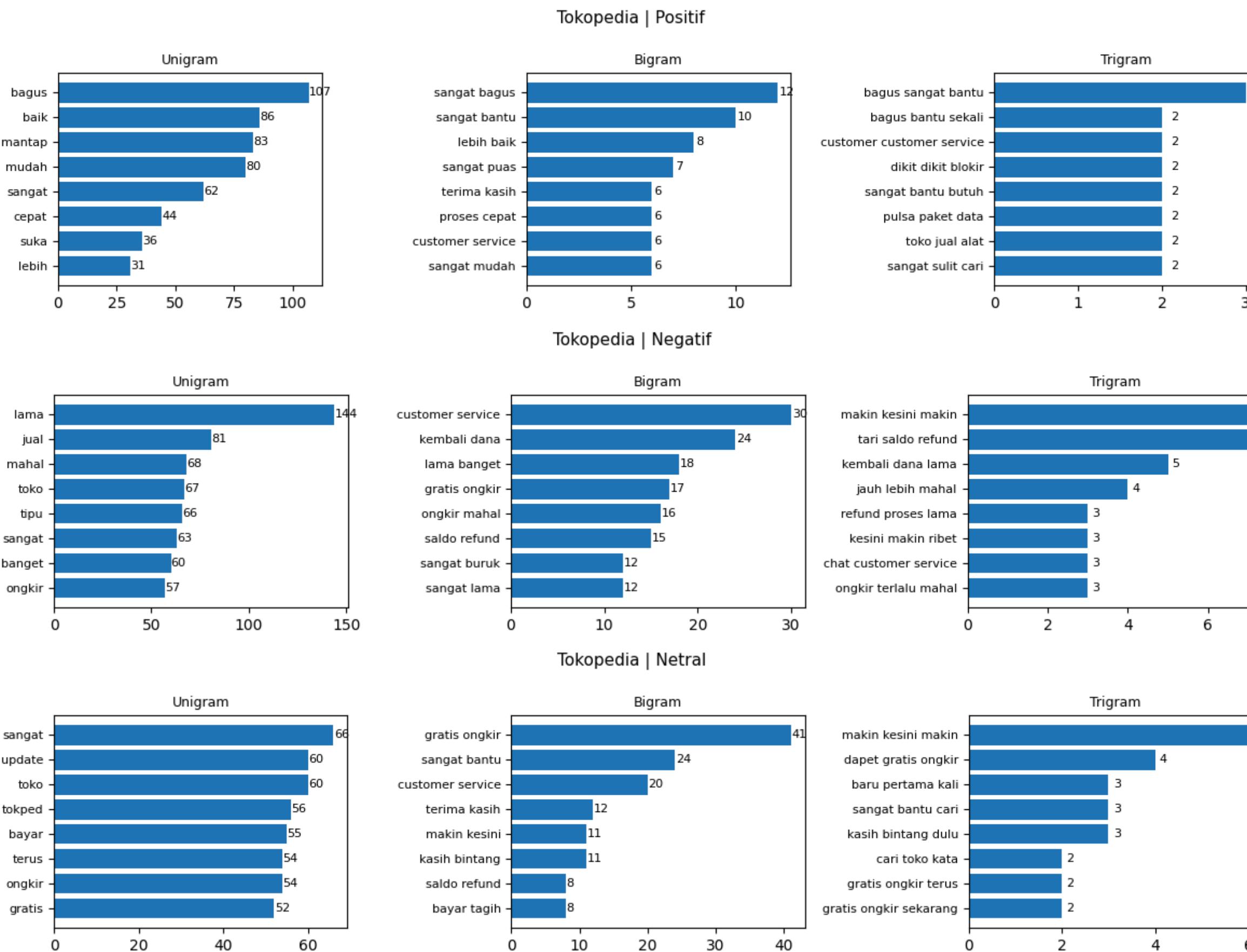
Neutra

banyak pesan udah ongkir

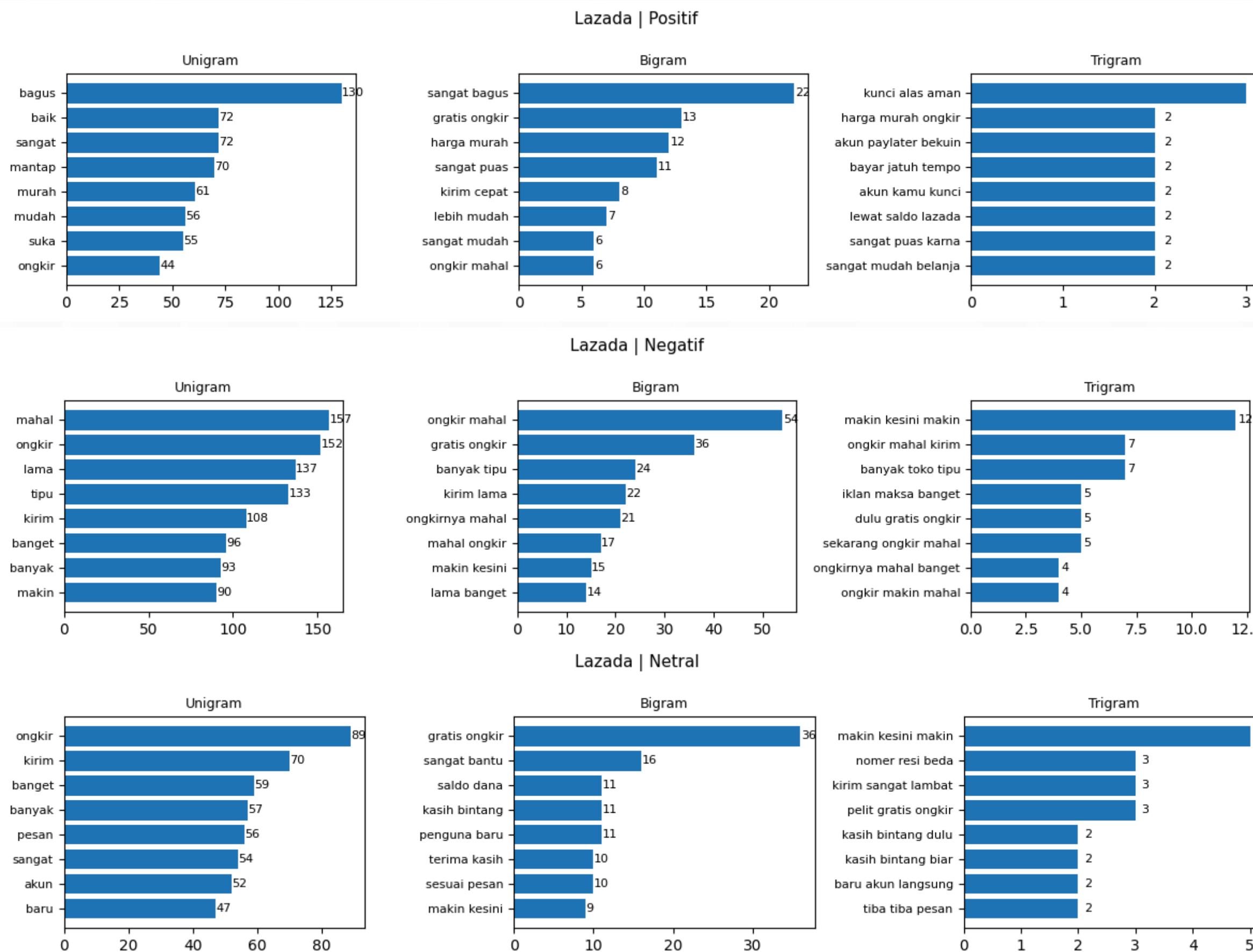
N-GRAM SENTIMEN MASING-MASING APLIKASI (SHOPEE)



N-GRAM SENTIMEN MASING-MASING APLIKASI (TOKOPEDIA)



N-GRAM SENTIMEN MASING-MASING APLIKASI (LAZADA)

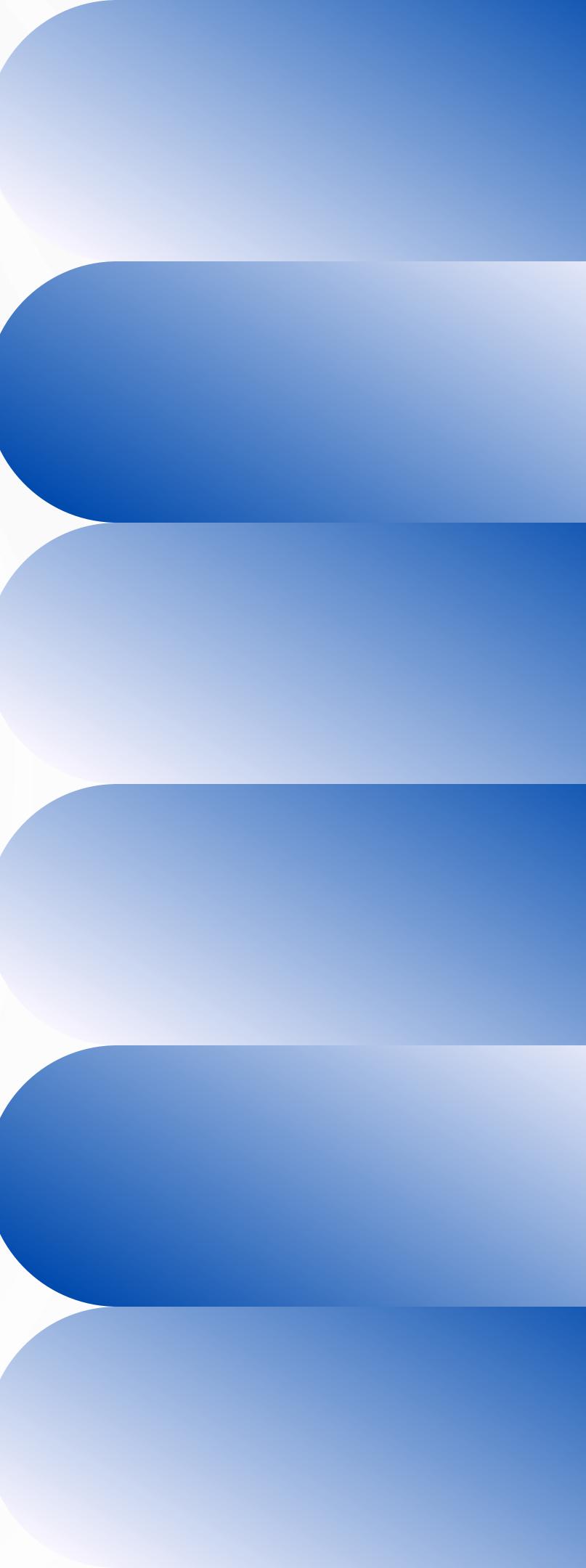


TABEL PERBANDINGAN SENTIMEN ULASAN PENGGUNA

Aspek Perbandingan	Shopee	Tokopedia	Lazada
Dominasi Sentimen	Netral	Netral	Netral
Sentimen Positif	Tinggi	Cukup tinggi	Paling rendah
Sentimen Negatif	Paling rendah	Sedang	Relatif tinggi
Kesan Umum Pengguna	Pengalaman pengguna memuaskan	Stabil dan fungsional	Pengalaman beragam
Kata Positif Dominan	bagus, murah, puas, mudah	bagus, mantap, cepat, mudah	bagus, murah, mantap, mudah
Fokus Keluhan Utama	ilikan, tiba-tiba, lama	lama, ongkir mahal, tipu	mahal, banyak tipu, lama
Karakteristik Ulasan Netral	Transaksi dan fitur aplikasi	Informasi layanan dan sistem	Proses belanja dan pengiriman

KESIMPULAN

- Shopee memiliki tingkat kepuasan pengguna tertinggi berdasarkan dominasi sentimen positif, yang menunjukkan bahwa sebagian pengguna merasa puas terhadap layanan yang diberikan. Hal ini didukung oleh kemunculan kata dan frasa positif pada hasil wordcloud dan n-gram, seperti kemudahan berbelanja, promo, serta transaksi yang praktis.
- Tokopedia menunjukkan distribusi sentimen yang lebih seimbang antara positif, negatif, dan netral. Memiliki ulasan yang beragam, mencakup pujian dan kritik terkait performa aplikasi dan layanan. Hal ini mencerminkan tingkat ekspektasi pengguna yang cukup tinggi terhadap aplikasi Tokopedia.
- Lazada memiliki proporsi sentimen negatif yang lebih menonjol dibandingkan dua aplikasi lainnya. Berdasarkan n-gram, sentimen negatif pada Lazada banyak berkaitan dengan permasalahan pengiriman, biaya ongkir, serta kualitas layanan pelanggan.



TERIMA KASIH

