

Web Scraping Dan Analisa Sentimen Ulasan Aplikasi SeaBank Menggunakan Deep Learning Long Short-Term Memory (LSTM)

Fitra Arya Ramadhanu

Desember 2025

Abstrak

Di era digitalisasi layanan perbankan, SeaBank hadir sebagai bank digital yang populer, memanfaatkan teknologi untuk memberikan kemudahan akses layanan keuangan. Aplikasi SeaBank berfungsi sebagai alat yang digunakan untuk meningkatkan kemudahan, kenyamanan, dan menarik minat nasabah melalui fitur-fitur serta promosi yang ditawarkan. Penelitian ini bertujuan untuk menganalisis sentimen pengguna aplikasi SeaBank menggunakan algoritma *Deep Learning Long Short-Term Memory (LSTM)*. *Dataset* yang digunakan adalah ulasan komentar pengguna di Google Play Store pada aplikasi SeaBank. Data diperoleh melalui metode web *scraping* untuk dijadikan sebagai objek analisis sentimen. Analisis ini penting untuk mengidentifikasi pandangan dan pengalaman pengguna secara kuantitatif. Hasil analisis sentimen yang didukung oleh visualisasi Word Cloud akan memberikan *insight* yang berharga bagi pihak SeaBank dalam memahami keunggulan (sentimen positif) dan mengidentifikasi area kekurangan (sentimen negatif) seperti isu teknis, fitur, atau kebijakan yang perlu dioptimalkan, sehingga dapat meningkatkan kualitas layanan dan kepuasan pengguna.

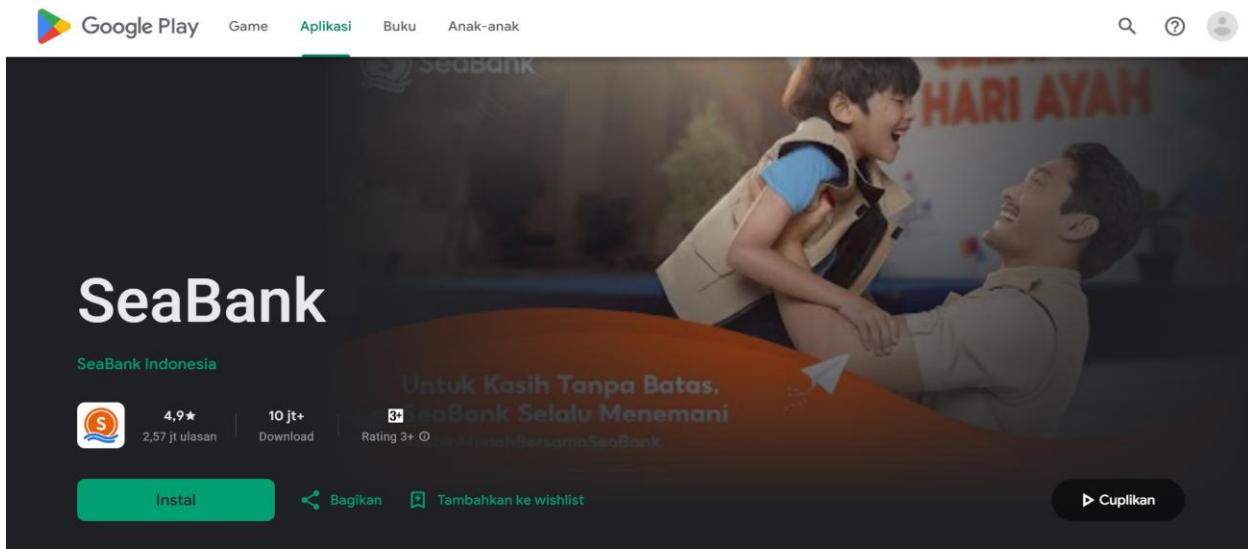
1. Tujuan

Tujuan pada penelitian ini adalah :

- Bagaimana LSTM dapat menganalisis sentimen terhadap ulasan komentar pada aplikasi SeaBank.
- Untuk memberikan insight kepada perusahaan SeaBank melalui WordCloud dalam peningkatan aplikasi SeaBank.

2. Dataset

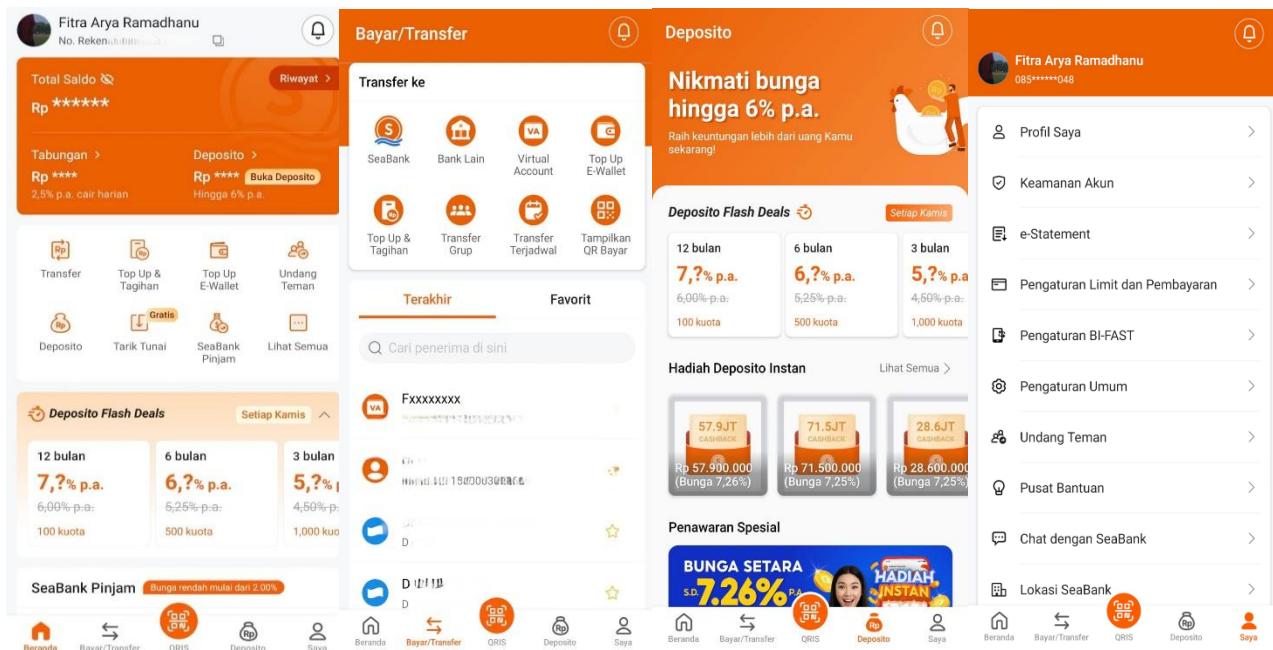
Dataset yang digunakan pada penelitian ini adalah ulasan komentar pada aplikasi SeaBank di App Store/Playstore. Data diambil melalui metode webscraping di Website ini :
https://play.google.com/store/apps/details?id=id.co.bankbkemobile.digitalbank&pcampaignid=web_share



Gambar 1: App SeaBank Di Google Play Store

3. Sea Bank

App SeaBank adalah aplikasi bank digital yang digunakan sebagai alat untuk meningkatkan kemudahan dan kenyamanan nasabah dalam mengakses layanan perbankan. Aplikasi SeaBank juga memiliki banyak fitur yang dapat digunakan oleh Pengguna seperti transfer, pembayaran, dan *top-up* e-wallet. Aplikasi SeaBank memanfaatkan pendekatan *artificial intelligence* (AI) untuk menciptakan sentuhan unik dan lebih modern dalam akses layanan keuangan, serta mendukung sistem keamanan yang baik sehingga aman untuk digunakan. Keunggulan SeaBank sering kali terletak pada penawaran bunga tabungan yang tinggi dan fasilitas transfer gratis.



Gambar 2 : User Interface pada App SeaBank

4. Metode

4.1 Text Mining

Text Mining merupakan proses penambangan data berupa teks yang bertujuan untuk mencari pola, informasi, dan pengetahuan yang berguna. Data teks yang digunakan umumnya berupa dokumen yang mewakili isi tertentu sehingga dapat dianalisa keterhubungannya. Dalam konteks penelitian ini, Text Mining menjadi tahapan krusial untuk mengolah data ulasan aplikasi SeaBank yang telah dikumpulkan melalui proses *Web Scraping*. Tujuan Text Mining di sini adalah untuk ekstraksi fitur yang relevan dari teks ulasan untuk mendukung proses Analisis Sentimen. Tahapan awal Text Mining yang dilakukan dalam penelitian ini adalah pra-pemrosesan data, yang sangat penting untuk memastikan data teks bersih dan siap digunakan sebagai *input* bagi model Deep Learning Long Short-Term Memory (LSTM).

4.2 Pre-Processing

Pada penelitian ini preprocessing dilakukan untuk mengelola data yang ada sehingga peneliti dapat menghindari gangguan pada data yang tidak konsisten. Tujuannya agar hasil output dari klasifikasi memiliki tingkat akurasi yang tinggi. Tahapan dari preprocessing meliputi casefolding, tokenizing, dan filtering. Adapun penjelasan dari masing masing tahapan tersebut antara lain:

a) Case Folding

Case Folding merupakan tahap mengubah semua huruf yang terdapat pada komentar menjadi huruf kecil. Hanya huruf dari “a” sampai “z” yang dapat di terima.

Teks Asli	Teks Setelah Case Folding
HI NAMA SAYA ARYA	hi nama saya arya
Aplikasi ini Bagus	aplikasi ini bagus.

b) Tokenizing

Tokenizing adalah proses memisahkan sebuah rangkaian teks atau *string* berdasarkan kata-kata yang menyusunnya, menjadi unit-unit yang lebih kecil yang disebut token. Secara garis besar memecah sekumpulan karakter dalam suatu teks kedalam satuan kata.

c) Filtering

Filtering adalah tahap mengambil kata-kata yang di anggap penting. Tahapan dalam filtering pada penelitian ini seperti:

- Menghapus angka tanda baca

- b) Menghapus Stopwords
- c) Normalisasi kata
- d) Stemming

4.3 Scraping

Scraping dilakukan untuk mengumpulkan data yang selanjutnya digunakan sebagai pembelajaran bagi Machine Learning. Alamat url aplikasi SeaBank di Google PlayStore.

ulasan	
...	0 mudah buat no rekening nya
1	praktis dan oke juga harga,terjangkau.programm...
2	ok
3	seabank pinjam ga di acc aneh haha,padahal sel...
4	kemudahan dalam transaksi dan transparan
...	...
4995	sipp
4996	sangat baik pengiriman nya juga lancar
4997	mantap no kendala,dan biaya admin pun gratis
4998	saya suka
4999	baguss

Gambar 3: Hasil Web Scraping

4.4 Labelisasi dengan Lexicon Based

Klasifikasi sentiment dengan Lexical Based adalah klasifikasi berdasarkan kata positif,negatif,ataupun netral yang ada pada ulasan komentar aplikasi SeaBank.

Klasifikasi dicocokkan dengan kata kata yang terdapat dalam kamus Lexicon Bahasa Indonesia, Jika komentar memiliki kata positif maka akan di golongkan pada sentiment positif, jika komentar memiliki kata negatif maka akan di golongkan pada sentiment negatif, namun jika komentar bernilai sama maka akan digolongkan pada sentiment netral.

Sentence Sentiment :

- o Positive if $S_{positive} > S_{negative}$
- o Neutral if $S_{positive} = S_{negative}$
- o Negative if $S_{positive} < S_{negative}$

	ulasan	clean_ulasan	sentimen
0	hati-hati guys aplikasinya gampang eror	hatihati guys gampang salah	neutra
1	sangat baik mudah akses ke bank mana pun apala...	sangat baik mudah akses bank mana sini gratis ...	positif
2	sangat membantu	sangat bantu	positif
3	udah gak jelas ada biaya adminnya tolol buka k...	gak jelas biaya adminnya tolol buka kecil	negatif
4	bagus	bagus	positif

Gambar 4: Hasil Labelling dengan Lexicon Based

4.5 Algoritma Deep Learning Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) adalah jenis arsitektur jaringan saraf tiruan yang termasuk dalam Recurrent Neural Network (RNN) yang di rancang untuk menangani masalah vanishing gradient dalam RNN konvensional.

a. Struktur LSTM

LSTM memiliki unit yang disebut sebagai sel memori, yang memungkinkan informasi dipertahankan dalam jangka waktu yang lebih lama dibandingkan RNN standar. Setiap sel LSTM memiliki tiga gerbang utama :

1. Forget Gate (Gerbang Lupa, f_t)

- Memutuskan informasi mana yang harus dihapus dari sel memori
- Jika suatu informasi tidak relevan, akan dihapus dari memori
- Menggunakan fungsi simoid untuk menghasilkan nilai antara 0 dan 1

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

a) Input Gate (Gerbang Input, i_t)

Memutuskan informasi baru yang akan ditambahkan ke sel memori. Memiliki dua bagian:

- Gerbang Input i_t : menentukan seberapa banyak informasi baru akan dimasukkan.
- Kandidat Memori

b) Memperbarui Sel Memori C_t

Menghasilkan informasi baru yang akan ditambahkan ke sel memori

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

c) Output Gate (Gerbang Output, O_t)

Memutuskan apa yang akan dihasilkan sebagai output dari unit LSTM.
Output dihitung menggunakan sel memori yang diperbarui.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

2. Memperbarui Sel Memori

Setelah gerbang forget dan input bekerja, sel memori diperbarui dengan :

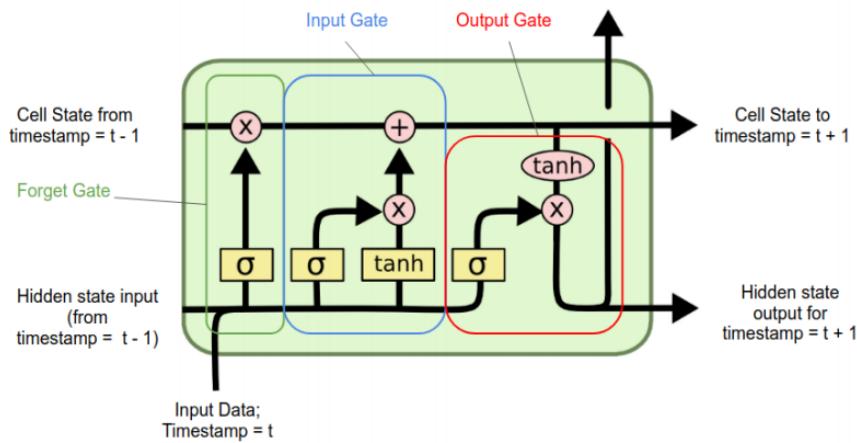
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

Lalu, output dari LSTM diberikan oleh :

$$h_t = o_t \odot \tanh(C_t)$$

Dimana:

- σ Sigma σ adalah fungsi aktivasi sigmoid.
- \tanh adalah fungsi aktivasi tangens hiperbolik.
- W dan b adalah bobot dan bias yang di pelajari oleh jaringan.



Hasil dan Pembahasan

a. Hasil Clean Ulasan

Pada tabel ulasan adalah teks yang belum di clean, tabel clean_ulasan adalah teks yang sudah di clean.

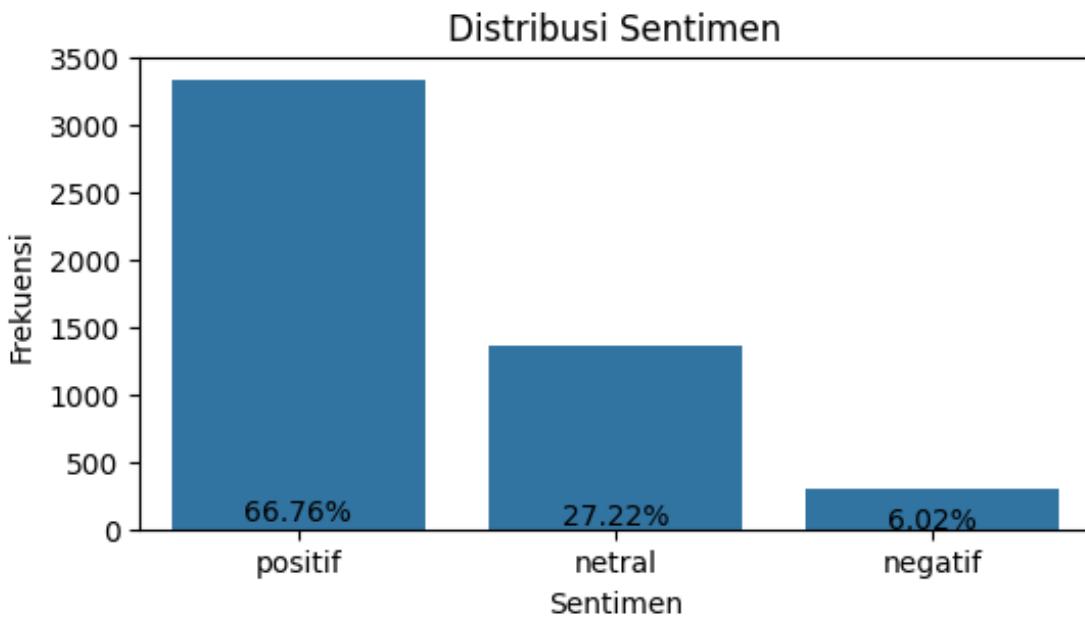
	ulasan	clean_ulasan
0	saya penguna aktif di seabank untuk keseluruha...	penguna aktif seabank seluruh seabank sangat b...
1	cepattt	cepattt
2	baik	baik
3	sangat membantu	sangat bantu
4	baik	baik

4.6. Visualisasi

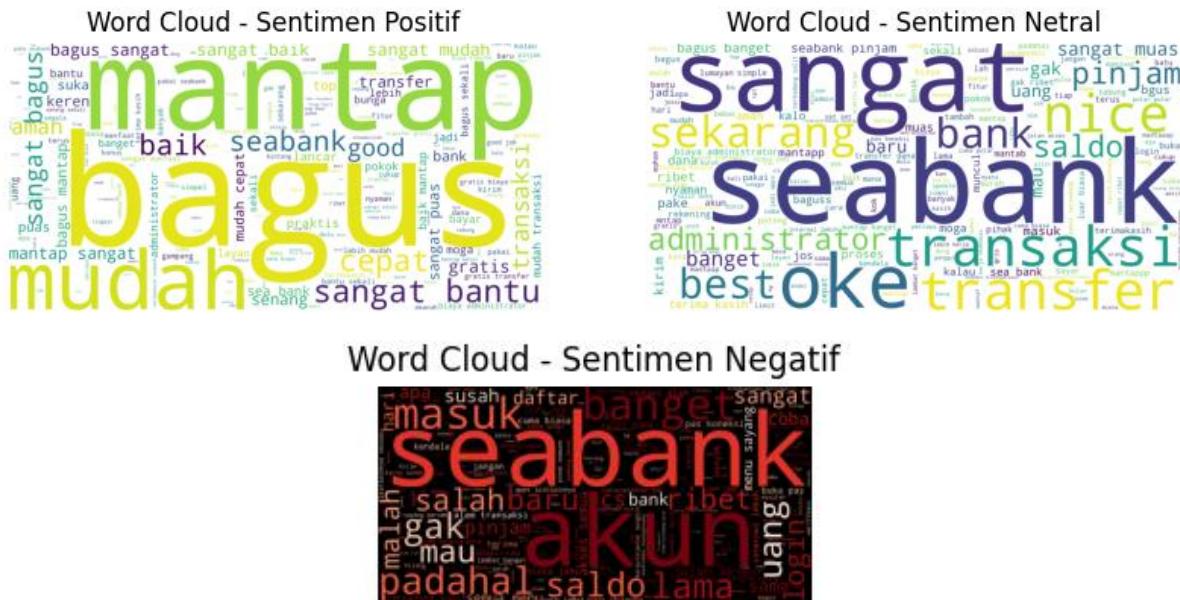
a) Persentase Sentimen

Setelah di labeling setiap teks menjadi positif, netral, dan negatif maka tahapan ini dilakukan perhitungan jumlah pada setiap sentimen.

Sentimen	Frekuensi	Persentase
Positif	3338	66.76
Netral	1361	27.22
Negatif	301	6.02



Jika dilihat hasil persentasenya maka sentimen terbanyak adalah sentimen positif kedua sentimen netral ketiga sentimen negatif. Untuk case ini kita perlu membuat visualisasi word cloud untuk melihat frekuensi kata yang paling banyak muncul.



- Pada sentimen Positif kata paling banyak muncul adalah Bagus, Mudah, Mantap, Cepat, Baik, Sangat.
- Pada sentimen Netral kata yang paling banyak muncul adalah Saya, Seabank, Sangat, Oke, Transaksi, Sekarang.
- Pada sentimen negatif kata yang paling banyak muncul adalah Seabank, Akun, Masuk, Susah, Login, Ribet, Lama.

4.7. Kesimpulan

Berdasarkan hasil analisis sentimen, ulasan dengan sentimen Positif sangat dominan dengan persentase sebesar 66,76% (3.338 data). Hal ini berbanding terbalik dengan sentimen negatif yang hanya sebesar 6,02% (301 data).

Kata-kata yang paling sering muncul dalam ulasan positif adalah "bagus", "mudah", dan "mantap". Hal ini mengindikasikan bahwa mayoritas pengguna merasa sangat puas dengan performa aplikasi yang ringan, proses transaksi yang cepat, serta antarmuka (UI) yang ramah pengguna.

Di sisi lain, meskipun jumlahnya sedikit, keluhan pada sentimen negatif berpusat pada kata kunci "masuk", "akun", "login", dan "susah". Ini menunjukkan bahwa kendala utama yang dialami sebagian kecil pengguna adalah masalah teknis saat mencoba mengakses aplikasi (gagal login) atau verifikasi akun yang dianggap rumit. Oleh karena itu, aspek yang perlu dipertahankan adalah kemudahan penggunaan (*usability*), sedangkan aspek yang harus segera diperbaiki adalah stabilitas sistem autentikasi/login.

Saran Perbaikan:

1. Optimasi Stabilitas Login dan Akses Akun

- Mengingat banyaknya keluhan dengan kata kunci "masuk", "login", dan "susah", pihak pengembang perlu melakukan audit teknis pada sistem autentikasi untuk meminimalkan kegagalan saat pengguna mencoba masuk ke aplikasi.
- Menyederhanakan proses verifikasi akun (jika ada indikasi kata "ribet" dan "lama") tanpa mengurangi standar keamanan, agar pengguna tidak kesulitan saat mendaftar atau memulihkan akun.

2. Peningkatan Responsivitas Layanan Pelanggan (CS)

- Munculnya kata "cs" dalam Word Cloud negatif mengindikasikan adanya ketidakpuasan terhadap layanan bantuan. Disarankan untuk meningkatkan kecepatan respons Customer Service dalam menangani keluhan pengguna yang mengalami kendala teknis (seperti saldo tertahan atau gagal transaksi).

3. Mempertahankan User Experience yang Ringan

- Kata mudah dan cepat pada sentimen positif adalah kekuatan utama SeaBank. Pengembangan fitur baru di masa depan harus dipastikan tidak memberatkan aplikasi, sehingga keunggulan kecepatan transaksi ini tetap terjaga.

```
!pip install google-play-scraper  
Requirement already satisfied: google-play-scraper in /usr/local/lib/python3.12/dist-packages (1.2.7)
```

```
!pip install wordCloud  
Requirement already satisfied: wordCloud in /usr/local/lib/python3.12/dist-packages (1.9.4)  
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.12/dist-packages (from wordCloud) (2.0.2)  
Requirement already satisfied: pillow in /usr/local/lib/python3.12/dist-packages (from wordCloud) (11.3.0)  
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from wordCloud) (3.10.0)  
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (1.3.3)  
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (0.12.1)  
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (4.61)  
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (1.4.  
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (25.0)  
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (3.2.5)  
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib->wordCloud) (2  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib->wor
```

```
!pip install Sastrawi  
Requirement already satisfied: Sastrawi in /usr/local/lib/python3.12/dist-packages (1.0.1)
```

```
import pandas as pd  
import numpy as np  
import re  
import matplotlib.pyplot as plt  
import seaborn as sns  
from wordcloud import WordCloud  
from google_play_scraper import reviews  
from tensorflow.keras.preprocessing.text import Tokenizer  
from tensorflow.keras.preprocessing.sequence import pad_sequences  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import classification_report, confusion_matrix  
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  
from tensorflow.keras.losses import sparse_categorical_crossentropy  
from google_play_scraper import reviews  
import string  
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
```

```
result, _ = reviews(  
    'id.co.bankbkemobile.digitalbank',  
    lang='id',  
    country='id',  
    count=5000  
)
```

```
df = pd.DataFrame(result)[['content']]  
df.rename(columns={'content':'ulasan'}, inplace=True)
```

```
df.head()
```

	ulasan	clean_ulasan	sentimen
0	saya penguna aktif di seabank untuk keseluruhan...	penguna aktif seabank seluruh seabank sangat b...	positif
1	cepattt	cepattt	netral
2	baik	baik	positif
3	sangat membantu	sangat bantu	positif
4	baik	baik	positif

```
stemmer = StemmerFactory().create_stemmer()
```

```
slang_dict = {
    # 1. Kata Ganti Orang / Sapaan
    "kmu": "kamu",
    "aq": "aku",
    "gw": "saya",
    "loe": "kamu",
    "lu": "kamu",
    "gua": "saya",
    "ane": "saya",
    "ente": "kamu",
    "gan": "juragan",
    "sis": "kakak perempuan",
    "sist": "kakak perempuan",
    "bro": "saudara laki-laki",
    "bos": "atasan",
    "min": "admin",

    "bgt": "banget",
    "gpp": "tidak apa-apa",
    "jd": "jadi",
    "gk": "tidak",
    "ga": "tidak",
    "ngga": "tidak",
    "tdk": "tidak",
    "jg": "juga",
    "trs": "terus",
    "dngn": "dengan",
    "sya": "saya",
    "bsa": "bisa",
    "skrg": "sekarang",
    "bkn": "bukan",
    "udh": "sudah",
    "sdh": "sudah",
    "tp": "tapi",
    "krn": "karena",
    "td": "tadi",
    "td": "tadi",
    "knp": "kenapa",
    "dpt": "dapat",
    "mnrt": "menurut",
    "dlm": "dalam",
    "smua": "semua",
    "trsbh": "tersebut",
    "ajh": "saja",
    "aja": "saja",
    "sj": "saja",
    "kl": "kalau",
    "klo": "kalau",
    "dr": "dari",
    "utk": "untuk",
    "bgmn": "bagaimana",
    "sm": "sama",
    "sampe": "sampai",
    "smpe": "sampai",
    "bgs": "bagus",
```

"apl": "aplikasi",
"apkasi": "aplikasi",
"fitur2": "fitur-fitur",
"trans": "transaksi",
"tf": "transfer",
"mbanking": "mobile banking",
"bank": "bank",
"rek": "rekening",
"ktp": "kartu tanda penduduk",
"akun": "akun",
"akunya": "akun",
"admin": "administrator",
"error": "kesalahan",
"eror": "kesalahan",
"lemot": "lambat",
"pending": "tertunda",
"verif": "verifikasi",
"kyk": "seperti",
"kyknya": "sepertinya",
"bagusss": "bagus",
"murahhh": "murah",
"mantul": "mantap betul",
"mrh": "marah",
"wkwk": "tertawa",
"lol": "tertawa terbahak-bahak",
"poka nya": "pokoknya",
"pokonya": "pokoknya",
"pkonya": "pokoknya",
"trmks": "terima kasih",
"smg": "semoga",
"yg" : "yang",

"baper": "terbawa perasaan",
"mager": "malas gerak",
"gabut": "tidak ada kerjaan",
"santuy": "santai",
"gaje": "tidak jelas",
"woles": "tenang",
"otw": "sedang dalam perjalanan",
"btw": "ngomong-ngomong",
"kuy": "ayo",
"cmiiw": "koreksi jika saya salah",
"asap": "secepat mungkin",
"japri": "jalur pribadi",
"rekber": "rekening bersama",
"bocil": "bocah kecil",
"vibes": "suasana",
"spill": "membocorkan",
"ghosting": "menghilang tanpa kabar",
"flexing": "pamer",
"cringe": "memalukan",
"ngegas": "marah",
"nolep": "tidak memiliki kehidupan sosial",
"halu": "halusinasi",
"brb": "akan segera kembali",
"pgn": "ingin",

```

    "akn": "akan",
    "bnr": "benar",
    "bener": "benar",
    "kmn": "kemana",
    "gmna": "bagaimana",
    "mkn": "makan",
    "mnum": "minum",
    "klr": "keluar",
    "bljr": "belajar",
    "trus": "terus",
    "bbrp": "beberapa",
    "trm ksh": "terima kasih",
    "trima kasih": "terima kasih",
    "mksh": "terima kasih",
    "makasih": "terima kasih",
    "gatau": "tidak tahu",
    "gatahu": "tidak tahu",
    "baper": "terbawa perasaan",
    "mager": "malas gerak",
    "murahhh": "murah",
    "mnatap": "mantap",
    "adm": "admin",
    "sangan": "sangat",
    "goof": "good",
    "smoga": "semoga",
    "trmksh": "terima kasih",
    "dgn": "dengan",
    "apk": "aplikasi",
    "ra": "tidak",
    "Ra": "tidak",
    "jls": "jelas",
    "gambar": "mudah",
    "nmr": "nomor",
    "klw": "kalau"
}

```

```

from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

# Inisialisasi Stop Word Remover
factory = StopWordRemoverFactory()
stopword_sastrawi = factory.get_stop_words()

# Tambahkan custom stop words
custom_stop_words = [
    "saya", "di", "ke", "dari", "yang", "dan", "atau", "adalah",
    "merupakan", "untuk", "buat", "itu", "ini", "dengan", "juga",
    "sudah", "udah", "aja", "nya", "aplikasi"
]

# Gabungkan stop words
all_stop_words = set(stopword_sastrawi)
all_stop_words.update(custom_stop_words)

def clean_text(text):
    text = text.lower() # Lowercasing
    text = re.sub(r'\d+', '', text) # Hapus angka
    text = text.translate(str.maketrans('', '', string.punctuation)) # Hapus tanda baca
    # Normalisasi Slang
    text = ' '.join([slang_dict[word] if word in slang_dict else word for word in text.split()])
    # Stemming
    text = stemmer.stem(text)
    # Penghapusan Stop Words
    text = ' '.join([word for word in text.split() if word not in all_stop_words])
    return text
}

df['clean_ulasan'] = df['ulasan'].apply(clean_text)

```

```

df.head()

      ulasan          clean_ulasan
0  saya penguna aktif di sebank untuk keseluruha...  penguna aktif sebank seluruh sebank sangat b...
1                  cepattt                      cepattt
2                   baik                         baik
3  sangat membantu                     sangat bantu
4                   baik                         baik

#Labeling Sentimen Berdasarkan Kata-kata Positif,Netral dan Negatif
positive_words = ["bagus", "mantap", "cepat", "puas", "senang", "baik", "suka", "gampang", "mudah", "lebih mudah", "serba mudah", "keren", "menarik", "membantu", "praktis", "top", "terbaik", "bantu", "puas", "memuaskan", "manfaat", "cocok", "lancar", "responsif", "efisien", "ok", "terjamin", "istimewa", "solusi", "hemat", "untung", "menguntungkan", "sukses", "berhasil", "recommended", "gratis", "free", "bunga", "tinggi", "cair", "bonus", "hadiah", "promosi", "promo", "mudah", "simpel", "fitur", " lengkap", "good", "terjangkau", "alhamdulil recommend ", "amanah", "aman"]

negative_words = ["buruk", "lama", "error", "jelek", "lelet", "kecewa", "penipuan", "blokir", "eror", "kurang", "berat", "bego", "lambat", "susah", "masalah", "banget", "gagal", "kendala", "sulit", "ribet", "lag", "hang", "crash", "rusak", "macet", "putus", "keluar", "tertunda", "pending", "payah", "menyebalkan", "sampah", "buang", "unir muter", "ngecas", "bobol", "hilang", "akun", "terkunci", "verifikasi", "gagal", "suspend", "diblokir", "pa tolong", "parah", "tolol", "salah", ]

netral_words = [ "debit", "kredit", "atm", "bank", "mandiri", "sinyal", "app", "mobile", "digital", "transfer", "tarik", "setor", "rekening", "saldo", "nominal", "jumlah", "transaksi", "dana", "cust "cs", "admin", "fitur", "menu", "layar", "login", "password", "pin", "otp", "email", "nomor", "telepon", "har upload", "download", "verifikasi", "proses", "sebank", "shopeepay", "shopee", "uang", "cuma", "hanya", "pa "kalau", "karena", "tersebut", "terbaru", "sekarang", ]

```



```

def sentimen_label(text):

    pos_count = sum(1 for word in text.split() if word in positive_words)
    netral_count = sum(1 for word in text.split() if word in netral_words)
    neg_count = sum(1 for word in text.split() if word in negative_words)

    if pos_count > neg_count:
        return 'positif'
    elif pos_count < neg_count:
        return 'negatif'
    else :
        return 'netral'

```



```

df ['sentimen'] = df['clean_ulasan'].apply (sentimen_label)

```



```

df.head()

```

	ulasan	clean_ulasan	sentimen
0	saya penguna aktif di sebank untuk keseluruha...	penguna aktif sebank seluruh sebank sangat b...	positif
1	cepattt	cepattt	netral
2	baik	baik	positif
3	sangat membantu	sangat bantu	positif
4	baik	baik	positif

```

df['sentimen'].value_counts()

   count
sentimen
positif    3338
netral     1361
negatif     301

dtype: int64

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
 0   ulasan      5000 non-null   object  
 1   clean_ulasan 5000 non-null   object  
 2   sentimen     5000 non-null   object  
dtypes: object(3)
memory usage: 117.3+ KB

#membuat visualisasi pada sentimen
sentimen_counts = df['sentimen'].value_counts()
sentimen_percentage = sentimen_counts / len(df) * 100
sentimen_df = pd.DataFrame({'Sentimen': sentimen_counts.index, 'Frekuensi': sentimen_counts.values, 'Persentase': sentimen_percentage})
print(sentimen_df)

plt.figure(figsize=(6, 3))
sns.barplot(x='Sentimen', y='Frekuensi', data=sentimen_df)
plt.title('Distribusi Sentimen')
plt.xlabel('Sentimen')
plt.ylabel('Frekuensi')

for i, v in enumerate(sentimen_percentage.values):
    plt.text(i, v + 5, f'{v:.2f}%', ha='center')

plt.show()

```

Sentimen	Frekuensi	Persentase
positif	3338	66.76
netral	1361	27.22
negatif	301	6.02

Distribusi Sentimen

	ulasan	clean_ulasan	sentimen
500	enk banget pake seabank	enk banget pake seabank	negatif
501	sungguh sangat membantu, tetap amanah	sungguh sangat bantu tetap amanah	positif
502	puasss bngettttt	puasss bngettttt	netral
503	mantap sanget enak nyaman aman	mantap sanget enak nyaman aman	positif
504	bestt	bestt	netral
...
4995	mantap banget	mantap banget	netral
4996	Saya ingin menutup rekening, tapi dipersulit, ...	tutup rekening sulit ganti nomor telepon sulit...	netral
4997	berguna.		netral
4998	seapinjam belom muncul muncul nih	seapinjam bom muncul muncul nih	netral
4999	ya sangat senang sekali dengan menggunakan sea...	sangat senang sekali seabank	positif

4500 rows × 3 columns

```
#konversi teks ke vektor (Tokenisasi & Padding)
tokenizer = Tokenizer()
tokenizer.fit_on_texts(df['clean_ulasan'])
X = tokenizer.texts_to_sequences(df['clean_ulasan'])
X = pad_sequences(X, maxlen=50) #Padding Sequences
```

```
list_label = []
for label in df['sentimen']:
    if label == 'positif':
        list_label.append(0)
    elif label == 'netral':
        list_label.append(1)
    else:
        list_label.append(2)
```

```
y=np.array(list_label)
```

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

```
model = Sequential([
    Embedding(input_dim=len(tokenizer.word_index)+1, output_dim=128, input_length=50),
    LSTM(128, return_sequences=True),
    Dropout(0.3),
    LSTM(64),
    Dense(32,activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax')
])
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/embedding.py:97: UserWarning: Argument `input_length` is deprecated
warnings.warn(
```

```
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy']))
```

```
history = model.fit(X_train, y_train, epochs=5, batch_size=16, validation_data=(X_test, y_test))
```

```
Epoch 1/5  
250/250 5s 11ms/step - accuracy: 0.6560 - loss: 0.7632 - val_accuracy: 0.9400 - val_loss: 0.1743  
Epoch 2/5  
250/250 2s 9ms/step - accuracy: 0.9447 - loss: 0.1661 - val_accuracy: 0.9690 - val_loss: 0.0845  
Epoch 3/5  
250/250 3s 12ms/step - accuracy: 0.9802 - loss: 0.0655 - val_accuracy: 0.9750 - val_loss: 0.0756  
Epoch 4/5  
250/250 2s 9ms/step - accuracy: 0.9906 - loss: 0.0354 - val_accuracy: 0.9700 - val_loss: 0.0837  
Epoch 5/5  
250/250 2s 10ms/step - accuracy: 0.9926 - loss: 0.0278 - val_accuracy: 0.9700 - val_loss: 0.0988
```

```
y_pred_probs = model.predict(X_test)
y_pred = np.argmax(y_pred_probs, axis=-1)
print(classification_report(y_test, y_pred))
```

32/32		0s 9ms/step			
	precision	recall	f1-score	support	
0	0.98	0.99	0.99	669	
1	0.94	0.95	0.95	268	
2	0.95	0.84	0.89	63	
accuracy				0.97	1000
macro avg		0.96	0.93	0.94	1000
weighted avg		0.97	0.97	0.97	1000

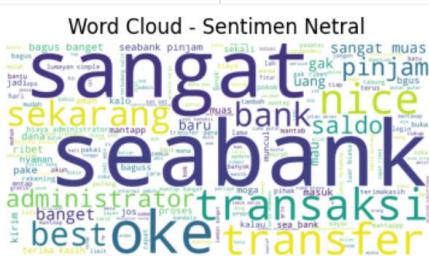
```
# Visualisasi Word Cloud (Positif & Negatif)
pos_text = ' '.join(df[df['sentimen'] == 'positif']['clean_ulasan'])
net_text = ' '.join(df[df['sentimen'] == 'neutraal']['clean_ulasan'])
neg_text = ' '.join(df[df['sentimen'] == 'negatif']['clean_ulasan'])

wordcloud_pos = WordCloud(width=2000, height=1000, background_color='white').generate(pos_text)
wordcloud_net = WordCloud(width=2000, height=1000, background_color='white').generate(net_text)
wordcloud_neg = WordCloud(width=2000, height=1000, background_color='black', colormap='Reds').generate(neg_text)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(wordcloud_pos)
plt.axis("off")
plt.title("Word Cloud - Sentimen Positif")

plt.subplot(1, 2, 2)
plt.imshow(wordcloud_net)
plt.axis("off")
plt.title("Word Cloud - Sentimen Neutraal")
plt.show()

plt.subplot(1, 2, 2)
plt.imshow(wordcloud_neg)
plt.axis("off")
plt.title("Word Cloud - Sentimen Negatif")
plt.show()
```



Word Cloud - Sentimen Negatif

