

LAPORAN PROJECT BASED LEARNING (PBL)
NATURAL LANGUAGE PROCESSING

**“ANALISIS SENTIMEN TERHADAP APLIKASI THREADS
MENGUNAKAN ALGORITMA K-MEANS CLUSTERING”**



Disusun Oleh:

KELOMPOK 6

Nazham Nabila	23/519393/NUGM/01133
Ichsanul Abid	23/519505/NUGM/01245
Alini Zaqira Kumiko	23/519557/NUGM/01296
Fairuz Hanifah	23/519562/NUGM/01301

DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
2023/2024

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT yang telah memberikan Rahmat dan inayah-Nya sehingga kami dapat menyelesaikan Laporan *Project Based Learning* (PBL) yang berjudul Analisis Sentimen Terhadap Aplikasi Threads Menggunakan Algoritma *K-Means Clustering*.

Terima kasih kami ucapkan kepada Ibu Prof. Dra. Sri Hartati, M.Sc., Ph.D. yang telah membantu kami baik secara moral maupun materi. Terima kasih juga kami ucapkan kepada teman-teman seperjuangan yang telah mendukung kami sehingga kami bisa menyelesaikan tugas ini tepat waktu.

Kami menyadari, bahwa Laporan *Project* yang kami buat ini masih jauh dari kata sempurna baik segi penyusunan, bahasa, maupun penulisannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pembaca guna menjadi acuan agar penulis bisa menjadi lebih baik lagi di masa mendatang.

Semoga Laporan *Project Based Learning* (PBL) ini bisa menambah wawasan para pembaca dan bisa bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan.

Yogyakarta, 9 Oktober 2023

Penulis

DESKRIPSI PEMBAGIAN TUGAS

1. Nazham Nabila (23/519393/NUGM/01133)
 - Melakukan visualisasi data dan memeriksa kualitas dari *dataset*, seperti memeriksa ada isi kolom yang kosong atau tidak.
 - Menyusun laporan akhir untuk Bab V Kesimpulan dan Saran.
2. Ichsanul Abid (23/519505/NUGM/01245)
 - Melakukan *preprocessing* data dari dataset teks yang telah diproses sebelumnya.
 - Menampilkan visualisasi hasil sentimen ulasan aplikasi berupa *Word Cloud*.
 - Menyusun laporan akhir untuk Bab II Analisis dan Perancangan.
 - Menyusun bahan presentasi dan finalisasi laporan akhir.
3. Alini Zaqira Kumiko (23/519557/NUGM/01296)
 - Melakukan pelatihan dan pengujian model sentimen dengan algoritma *K-Means Clustering*.
 - Memberikan *comment* pada *source code* program.
 - Menyusun laporan akhir untuk Bab III Implementasi serta Bab IV Hasil dan Pembahasan.
4. Fairuz Hanifah (23/519562/NUGM/01301)
 - Mengolah data mentah dan melakukan *translating* serta *trimming dataset*.
 - Menyusun laporan akhir untuk Bab I Deskripsi Tugas serta Bab IV Hasil dan Pembahasan.

DAFTAR ISI

KATA PENGANTAR	ii
DESKRIPSI PEMBAGIAN TUGAS	iii
DAFTAR ISI	iv
BAB I DESKRIPSI TUGAS	1
1.1 LATAR BELAKANG	1
1.2 RUMUSAN MASALAH.....	2
1.3 TUJUAN	2
BAB II ANALISIS DAN PERANCANGAN.....	3
2.1 METODE YANG DIGUNAKAN.....	3
2.2 PERANCANGAN MANUAL	4
2.3 PERANCANGAN PROSES	5
2.4 PERANCANGAN MODUL.....	9
BAB III IMPLEMENTASI.....	10
3.1 SOURCE CODE DAN COMMENT	10
3.2 SUMBER SOURCE CODE.....	15
3.3 DATASET	16
BAB IV HASIL DAN PEMBAHASAN	18
4.1 PENGUJIAN.....	18
4.2 PEMBAHASAN	20
BAB V PENUTUP.....	23
5.1 KESIMPULAN	23
5.2 SARAN	23
DAFTAR PUSTAKA.....	v

BAB I

DESKRIPSI TUGAS

1.1 LATAR BELAKANG

Dalam era digital yang berkembang pesat saat ini, aplikasi media sosial telah menjadi bagian yang tak terpisahkan dalam kehidupan sehari-hari kita. Salah satu aplikasi *mobile* yang menonjol dalam kategori ini adalah Threads. Dengan jutaan pengguna aktif di seluruh dunia, Threads telah menjadi sorotan dalam dunia media sosial. Untuk memahami lebih baik bagaimana aplikasi ini diterima oleh penggunanya, analisis sentimen menjadi suatu kebutuhan yang tidak dapat diabaikan.

Aplikasi *mobile* semakin merajalela dengan kemajuan teknologi yang terus berlanjut. Threads, sebagai salah satu aplikasi *mobile* yang populer, telah menerima banyak ulasan dan umpan balik dari penggunanya. Mengetahui apa yang dipikirkan dan dirasakan oleh pengguna aplikasi ini melalui ulasan-ulasan mereka menjadi sangat penting. Oleh karena itu, penting untuk menjalankan analisis sentimen terhadap ulasan-ulasan tersebut agar kita dapat mendapatkan wawasan yang lebih mendalam tentang bagaimana pengguna merespon aplikasi Threads.

Analisis sentimen ini tidak hanya bermanfaat bagi pengguna, tetapi juga bagi pengembang aplikasi. Dengan memahami sentimen positif, negatif, dan netral yang terkandung dalam ulasan-ulasan pengguna, pengembang dapat mengidentifikasi kekuatan dan kelemahan dari aplikasi ini. Hal ini membantu mereka untuk mengambil tindakan yang lebih tepat dalam meningkatkan pengalaman pengguna secara keseluruhan. Selain itu, dengan memahami bagaimana ulasan pengguna mempengaruhi citra dan popularitas aplikasi di *platform online*, pengembang dapat merencanakan strategi pengembangan dan pemasaran yang lebih efektif.

Dalam dunia yang terhubung secara digital seperti saat ini, opini pengguna di *platform online* memiliki dampak besar pada citra dan popularitas suatu produk atau layanan. Oleh karena itu, pemahaman yang mendalam tentang sentimen yang terkandung dalam ulasan-ulasan pengguna adalah kunci untuk membentuk strategi pengembangan dan pemasaran yang efektif. Pendekatan seperti *K-Means Clustering* dapat menjadi alat yang sangat berguna dalam menganalisis sentimen ini dan membantu pengembang dan pemangku kepentingan lainnya dalam mengambil keputusan yang terinformasi untuk mengembangkan dan memperbaiki aplikasi Threads.

1.2 RUMUSAN MASALAH

Adapun rumusan masalah dalam penelitian kami yaitu.

1. Bagaimana cara menganalisis sentimen positif, negatif, dan netral dari ulasan pengguna terhadap aplikasi Threads?
2. Bagaimana menerapkan algoritma *K-Means Clustering* untuk mengelompokkan ulasan-ulasan pengguna menjadi kelompok-kelompok dengan sentimen yang serupa?
3. Bagaimana hasil analisis sentimen dan pengelompokkan ini dapat memberikan wawasan yang berguna bagi pengembang aplikasi Threads untuk perbaikan lebih lanjut?

1.3 TUJUAN

Tujuan penelitian kami yaitu.

1. Menganalisis sentimen dari ulasan pengguna terhadap aplikasi Threads.
2. Mengimplementasikan algoritma *K-Means Clustering* untuk mengelompokkan ulasan-ulasan pengguna berdasarkan sentimennya.
3. Menyajikan hasil analisis sentimen dan pengelompokkan sebagai wawasan bagi pengembang aplikasi Threads untuk melakukan perbaikan dan peningkatan yang lebih baik.

BAB II

ANALISIS DAN PERANCANGAN

2.1 METODE YANG DIGUNAKAN

Analisis sentimen ulasan aplikasi Threads adalah proses untuk mengidentifikasi dan mengklasifikasikan opini, sentimen, atau emosi pengguna terhadap aplikasi Threads. Algoritma *K-Means Clustering* adalah salah satu metode yang dapat digunakan dalam analisis sentimen aplikasi ini. Algoritma ini merupakan salah satu metode pengelompokan data yang tidak berlabel (*unsupervised learning*) berdasarkan kemiripan fitur-fitur yang dimiliki. Algoritma ini bekerja dengan cara menentukan sejumlah k sebagai jumlah *cluster* yang diinginkan, kemudian memilih secara acak k titik data sebagai pusat *cluster* awal (*centroid*). Selanjutnya, algoritma ini menghitung jarak setiap data terhadap *centroid* menggunakan rumus *Euclidean Distance*, yaitu:

$$d(x_i, \mu_j) = \sqrt{\sum_{k=1}^n (x_{ik} - \mu_{jk})^2}$$

Keterangan:

$d(x_i, \mu_j)$ = Jarak antara data ke- i dan *centroid* ke- j

x_i = Data ke- i , dimana $i = 1, 2, \dots, m$

μ_j = *Centroid* ke- j , dimana $j = 1, 2, \dots, k$

n = Jumlah fitur dari data

Data yang memiliki jarak terdekat dengan suatu *centroid* akan menjadi anggota dari *cluster* yang dipimpin oleh *centroid* tersebut. Setelah semua data dikelompokkan, algoritma ini akan menghitung ulang nilai *centroid* berdasarkan rata-rata dari anggota *cluster*. Proses ini akan diulangi sampai nilai *centroid* tidak berubah lagi atau mencapai batas iterasi yang ditentukan.

Algoritma *K-Means Clustering* dapat digunakan untuk melakukan pemodelan topik pada data ulasan aplikasi Threads. Pemodelan topik adalah proses untuk menemukan topik-topik yang muncul dalam data ulasan berdasarkan distribusi kata-kata yang sering muncul bersama-sama. Dengan menggunakan algoritma *K-Means Clustering*, kita dapat mengelompokkan data ulasan ke dalam 3 *cluster* yang merepresentasikan sentimen ulasan aplikasi, yaitu “Positif” untuk ulasan pengguna yang mungkin mengungkapkan kepuasan atau kegembiraan dengan aplikasi Threads, “Negatif” untuk ulasan pengguna yang

mungkin memiliki keluhan atau ketidakpuasan terhadap beberapa aspek aplikasi Threads, dan “Netral” untuk ulasan pengguna yang tidak mengandung evaluasi sentimen yang kuat.

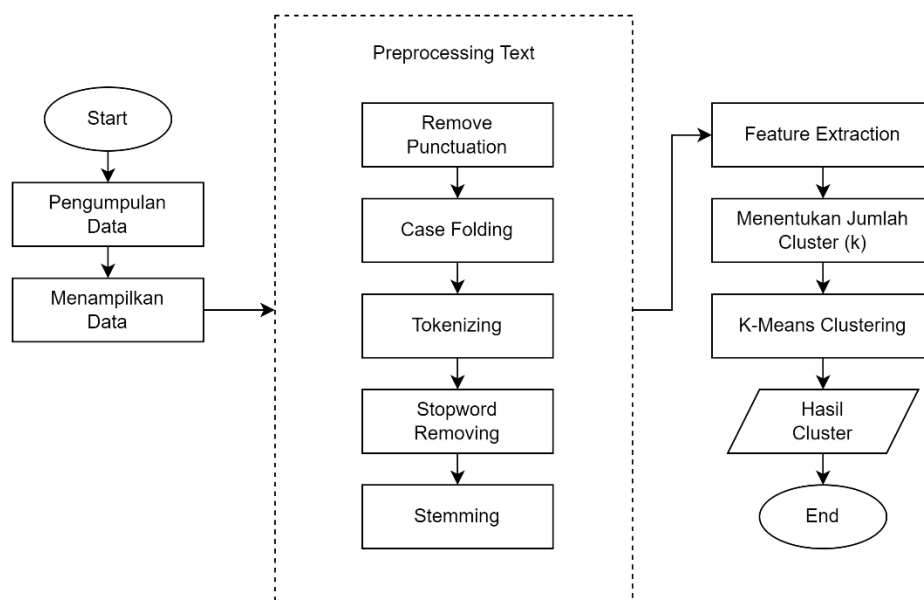
2.2 PERANCANGAN MANUAL

Perancangan manual sebuah sistem yang menganalisis sentimen ulasan aplikasi Threads menggunakan algoritma *K-Means Clustering* terdiri dari beberapa langkah, yaitu sebagai berikut.

1. Menentukan tujuan dan ruang lingkup sistem. Tujuan sistem adalah untuk mengelompokkan ulasan pengguna aplikasi Threads ke dalam beberapa kategori sentimen, seperti positif, negatif, atau netral. Ruang lingkup sistem meliputi sumber data, metode analisis, dan hasil analisis.
2. Mengumpulkan data ulasan pengguna aplikasi Threads dari situs penyedia aplikasi, seperti Google Play Store atau App Store. Data ulasan pengguna aplikasi berupa teks yang mengandung pendapat atau evaluasi pengguna terhadap aplikasi yang digunakan.
3. Melakukan *preprocessing* data, yaitu membersihkan teks dari karakter yang tidak relevan, mengubah teks menjadi huruf kecil, membuat token-token dari teks, menghapus kata-kata yang tidak berarti (*stopwords*), dan mengubah kata ke dalam bentuk dasarnya.
4. Melakukan ekstraksi fitur, yaitu mengubah teks menjadi vektor numerik yang merepresentasikan bobot kata-kata dalam teks.
5. Menentukan jumlah *cluster* (k) yang optimal, dimana akan diinisialisasi nilai $k=3$ yang merepresentasikan sentimen positif, negatif, dan netral untuk setiap ulasan yang dianalisis.
6. Melakukan *K-Means Clustering* dengan menggunakan nilai k yang optimal. Algoritma ini akan menginisialisasi *centroid* secara acak, kemudian menghitung jarak antara setiap data dan *centroid*, dan menempatkan data ke *cluster* terdekat. Selanjutnya, algoritma ini akan menghitung ulang *centroid* berdasarkan rata-rata data dalam *cluster*, dan mengulangi proses sebelumnya sampai *centroid* tidak berubah lagi atau mencapai iterasi maksimum.
7. Memberikan label sentimen untuk setiap *cluster* dengan menggunakan metode mayoritas. Metode ini akan menghitung frekuensi kata-kata positif dan negatif dalam setiap *cluster*, dan memberikan label sesuai dengan kata-kata yang paling dominan. Jika frekuensi kata-kata positif dan negatif sama, maka labelnya adalah netral.

8. Menampilkan hasil analisis sentimen ulasan pengguna aplikasi dalam bentuk tabel atau grafik yang menunjukkan distribusi atau proporsi ulasan berdasarkan kategori sentimen.

2.3 PERANCANGAN PROSES



Gambar 1.1 Rancangan Sistem

Analisis sentimen ulasan aplikasi Threads adalah proses untuk mengidentifikasi dan mengklasifikasikan opini, sentimen, atau emosi pengguna terhadap aplikasi Threads. Untuk melakukan analisis sentimen ini, kita dapat menggunakan beberapa langkah sebagai berikut.

1. Pengumpulan Data

Tahap pengumpulan data untuk analisis sentimen terhadap aplikasi Threads dengan menggunakan algoritma *K-Means Clustering* dimulai dengan identifikasi sumber data yang relevan. Pertama, mengidentifikasi sumber-sumber yang menyediakan ulasan dan sentimen terkait aplikasi Threads. Sumber data ini dapat berasal dari *platform-platform* ulasan pengguna, jejaring sosial, forum diskusi, atau situs web yang memungkinkan pengguna untuk memberikan ulasan terkait aplikasi Threads.

Selanjutnya, setelah sumber data teridentifikasi, langkah kedua adalah memilih dan mengakses sumber data tersebut. Pemilihan dilakukan sesuai dengan kebutuhan dan relevansi dengan aplikasi Threads yang akan dianalisis.

Langkah berikutnya adalah melakukan pengumpulan data ulasan dari sumber yang telah dipilih. Data yang dikumpulkan harus mencakup ulasan atau komentar pengguna terkait dengan aplikasi Threads tersebut. Penting untuk mengumpulkan data dengan cermat dan memastikan data yang terkumpul cukup representatif dan bervariasi.

Setelah data ulasan terkumpul, tahap selanjutnya adalah memastikan data tersebut tersimpan dalam format yang sesuai untuk analisis lebih lanjut. Data dapat disimpan dalam berbagai format, seperti teks, CSV, atau basis data, sesuai dengan preferensi dan kebutuhan analisis.

Terakhir, sebelum memasuki tahap analisis sentimen menggunakan algoritma *K-Means Clustering*, diperlukan pembersihan dan *preprocessing* data. Hal ini mencakup membersihkan data dari informasi yang tidak relevan, menghilangkan duplikat, melakukan normalisasi teks, penghapusan tanda baca, dan pengaturan ejaan untuk memastikan data siap digunakan dalam proses analisis sentimen. Dengan tahapan ini, data siap untuk direpresentasikan sebagai fitur dan diteruskan ke algoritma *K-Means Clustering* untuk mengelompokkan sentimen berdasarkan kesamaan fitur.

2. *Preprocessing Data*

Preprocessing data adalah langkah kritis dalam analisis sentimen terhadap aplikasi Threads menggunakan algoritma *K-Means Clustering*. Tahapan ini membantu mempersiapkan teks yang akan dianalisis agar lebih mudah dipahami oleh komputer. *Preprocessing* meliputi beberapa teknik yaitu sebagai berikut.

a. *Remove Punctuation*

Tahap ini adalah untuk menghapus tanda baca seperti titik, koma, tanda tanya, dan lain-lain dari data teks. Tanda baca biasanya tidak memberikan informasi yang berguna untuk analisis teks dan dapat mengganggu proses tokenisasi dan *stemming*. Contoh: “Aplikasi ini cukup bagus dan memuaskan.” → “Aplikasi ini cukup bagus dan memuaskan”

b. *Case Folding*

Tahap ini adalah untuk menyeragamkan huruf besar dan huruf kecil dalam data teks. Huruf besar dan huruf kecil dapat membedakan makna kata dalam bahasa tertentu, tetapi dalam analisis teks, kita biasanya tidak memperhatikan perbedaan tersebut. Contoh: “Aplikasi ini cukup bagus dan memuaskan” → “aplikasi ini cukup bagus dan memuaskan”

c. *Tokenizing*

Tahap ini adalah untuk memecah data teks menjadi unit-unit yang lebih kecil yang disebut token. Token biasanya berupa kata, tetapi dapat juga berupa frasa atau karakter. Tokenisasi berguna untuk mengidentifikasi kata-kata yang menjadi fitur dalam analisis teks. Contoh: “aplikasi ini cukup bagus dan memuaskan” → [“aplikasi”, “ini”, “cukup”, “bagus”, “dan”, “memuaskan”]

d. *Stopword Removing*

Tahap ini adalah untuk menghilangkan kata-kata yang sering muncul dalam data teks tetapi tidak memberikan informasi yang penting untuk analisis teks. Kata-kata ini disebut *stopwords* dan biasanya berupa kata sambung, kata ganti, atau preposisi. *Stopwords* dapat berbeda-beda tergantung pada bahasa dan domain data teks. Contoh: [“aplikasi”, “ini”, “cukup”, “bagus”, “dan”, “memuaskan”] → [“aplikasi”, “cukup”, “bagus”, “memuaskan”]

e. *Stemming*

Tahap ini adalah untuk mengubah kata-kata yang memiliki imbuhan atau bentuk tidak beraturan menjadi bentuk dasarnya atau akarnya. *Stemming* berguna untuk mengurangi variasi kata yang memiliki makna yang sama atau serupa dalam analisis teks. *Stemming* memerlukan aturan-aturan yang sesuai dengan kaidah bahasa yang digunakan. Contoh: [“aplikasi”, “cukup”, “bagus”, “memuaskan”] → [“aplikasi”, “cukup”, “bagus”, “puas”]

3. *Feature Extraction*

Setelah data diproses, akan dilakukan *feature extraction* pada data teks untuk mengubah teks menjadi vektor numerik yang dapat diproses oleh algoritma *machine learning K-Means Clustering*. *Feature extraction* dilakukan dengan menggunakan metode *word embeddings* (Word2Vec) untuk menghasilkan representasi vektor numerik pada setiap ulasan.

4. Menentukan Jumlah *Cluster* (k)

Menentukan jumlah *cluster* (k) adalah langkah penting dalam perancangan analisis sentimen menggunakan algoritma *K-Means Clustering*. Dalam konteks ini, kita telah menentukan k=3, yang berarti kita akan menggunakan tiga *cluster* berbeda untuk mengelompokkan ulasan pengguna aplikasi Threads, yaitu “Positif” untuk ulasan pengguna yang mungkin mengungkapkan kepuasan atau kegembiraan dengan aplikasi Threads, “Negatif” untuk ulasan pengguna yang mungkin memiliki keluhan

atau ketidakpuasan terhadap beberapa aspek aplikasi Threads, dan “Netral” untuk ulasan pengguna yang tidak mengandung evaluasi sentimen yang kuat.

5. *K-Means Clustering*

Tahap selanjutnya adalah menginisialisasi pusat *cluster* sebagai titik-titik awal yang akan digunakan dalam algoritma *K-Means Clustering* dan melakukan iterasi *K-Means*, dimana setiap ulasan diberikan kepada *cluster* yang memiliki pusat terdekat berdasarkan jarak Euclidean antara vektor representasi ulasan dan pusat *cluster*. Kemudian, pusat *cluster* diperbarui berdasarkan rata-rata dari semua vektor dalam *cluster* tersebut. Iterasi ini akan terus berlanjut hingga tidak ada perubahan dalam pengelompokan ulasan atau perubahan dalam pusat *cluster*. Ini menandakan konvergensi algoritma.

6. Hasil *Cluster*

Menampilkan hasil *clustering* setelah konvergensi tercapai dengan menghasilkan kelompok-kelompok ulasan berdasarkan sentimen positif, negatif, dan netral. Kita dapat menganalisis ulasan-ulasan dalam masing-masing *cluster* untuk memahami pola-pola sentimen.

7. Evaluasi *Cluster*

Tahap evaluasi *cluster* dalam analisis sentimen merupakan langkah penting untuk mengukur keefektifan dan validitas hasil klasterisasi yang telah dilakukan. Hasil evaluasi *cluster* ini sangat penting karena membantu dalam menentukan apakah jumlah *cluster* yang dipilih awalnya sesuai dengan data dan apakah *cluster-cluster* tersebut membentuk kelompok sentimen yang bermakna.

Jika evaluasi mengungkapkan masalah, seperti adanya *cluster* yang tumpang tindih atau data yang tidak terdistribusi dengan baik, maka dapat dilakukan penyesuaian pada jumlah *cluster* atau teknik analisis lainnya. Dengan demikian, tahap evaluasi *cluster* menjadi langkah kunci dalam memastikan bahwa hasil analisis sentimen menggunakan algoritma *K-Means Clustering* dapat memberikan wawasan yang akurat dan berguna terkait dengan ulasan pengguna aplikasi Threads.

8. Analisis dan Interpretasi Hasil

Setelah *cluster* sentimen terbentuk dan evaluasi telah dilakukan, dapat dipahami bagaimana ulasan pengguna aplikasi Threads terbagi ke dalam kelompok-kelompok sentimen yang berbeda. Misalnya, hasil analisis mungkin mengungkapkan bahwa terdapat tiga kelompok sentimen utama terkait aplikasi ini: positif, negatif dan netral. Kelompok sentimen positif akan berisi ulasan-ulasan yang mencerminkan

pengalaman positif pengguna dengan aplikasi Threads, sementara kelompok sentimen negatif akan mencakup ulasan-ulasan yang mencatat masalah atau ketidakpuasan pengguna. Kelompok sentimen netral mungkin berisi ulasan-ulasan yang tidak mengungkapkan sentimen kuat, atau berisi informasi yang lebih objektif. Selanjutnya, analisis akan mengidentifikasi kata-kata kunci atau topik-topik yang mendominasi setiap kelompok sentimen. Dengan demikian, analisis dan interpretasi hasil dari analisis sentimen ini bukan hanya memberikan pemahaman tentang apa yang dipikirkan pengguna, tetapi juga menjadi dasar untuk pengambilan keputusan yang lebih baik dalam pengembangan dan manajemen aplikasi Threads.

2.4 PERANCANGAN MODUL

Modul Analisis Sentimen (*Input: Dataset Ulasan, Output: Hasil Sentimen*)

Input:

1. *Dataset Ulasan* → *Input* pertama adalah *dataset* yang berisi ulasan pengguna tentang aplikasi Threads. *Dataset* ini berisi kolom-kolom seperti “review_description”, “rating”, “review_date”, dan “translated_description”. *Dataset* ini digunakan sebagai sumber data untuk analisis sentimen.

Output:

1. Hasil Sentimen → *Output* utama dari modul ini adalah hasil analisis sentimen untuk setiap ulasan dalam *dataset*. Hasil ini berupa label sentimen, seperti “Positif”, “Negatif”, atau “Netral”, yang diberikan kepada setiap ulasan berdasarkan *clustering* dengan algoritma *K-Means*.
2. Visualisasi Hasil Sentimen → *Output* juga dapat berupa visualisasi hasil sentimen dalam bentuk grafik atau *Word Cloud*, yang membantu dalam memahami kata-kata kunci yang sering digunakan dalam setiap kategori sentimen.
3. Label *Cluster* → *Output* berupa label *cluster* untuk setiap ulasan yang dapat digunakan untuk analisis lebih lanjut atau pengelompokkan ulasan berdasarkan sentimen tertentu.

BAB III

IMPLEMENTASI

3.1 SOURCE CODE DAN COMMENT

1. Mempersiapkan Program

```
from google.colab import drive
# Mengimpor modul drive dari google.colab
drive.mount('/content/drive')
# Mount Google Drive ke direktori /content/drive
!pip install Sastrawi
# Menggunakan perintah pip untuk menginstal pustaka Sastrawi
!pip install joblib
# Menggunakan perintah pip untuk menginstal pustaka joblib
```

Kode di atas digunakan dalam lingkungan Google Colab untuk melakukan tiga tugas persiapan. Pertama, kode mengimpor modul “drive” dan melakukan mounting Google Drive ke direktori “/content/drive”, memungkinkan akses ke file di Google Drive. Kedua, kode menginstal pustaka Sastrawi, yang digunakan untuk pemrosesan bahasa Indonesia, termasuk stemming. Terakhir, pustaka joblib diinstal untuk menyimpan dan memuat objek Python yang lebih besar seperti model mesin pembelajaran. Kode ini membantu mempersiapkan lingkungan untuk analisis teks atau pemrosesan bahasa alami dalam proyek yang berjalan di Google Colab.

2. Menampilkan Data

```
import pandas as pd
# Mengimpor pustaka pandas untuk manipulasi data
import numpy as np
# Mengimpor pustaka numpy untuk operasi numerik

pd.set_option('display.max_columns', None)
df = pd.read_csv('/content/drive/MyDrive/nlp-project/5000-Data.csv')
df.head(50)
# Memanggil dataset dan menampilkan 50 data teratas untuk preview

df.info()
# Menampilkan informasi tentang dataset seperti jumlah baris,
jumlah kolom, nama kolom, jumlah nilai non-null, tipe data kolom,
dan jumlah memori yang digunakan.

df.isna()
#Memeriksa setiap baris yang memiliki nilai null pada semua kolom
dataset dengan menggunakan sintaks isna() yang disediakan library
pandas.

df.isna().any()
#Memeriksa apakah ada nilai yang hilang dalam setiap kolom
dataset.

df.describe()
```

```
#Menampilkan statistik deskriptif tentang dataset seperti jumlah,
mean, standar deviasi, nilai minimum, kuartil 1, median, kuartil
3, dan nilai maksimum dari kolom Skor Kepuasan.

df.isnull().sum()
#Mencari jumlah baris data yang bernilai null dan memeriksa apakah
terdapat kolom label yang memiliki nilai kosong.
```

Kode di atas merupakan langkah awal dalam analisis data menggunakan pustaka pandas dan numpy, yang merupakan komponen penting dalam ilmu data dan analisis data. Pertama, kita mengimpor pustaka pandas untuk manipulasi data dan pustaka numpy untuk operasi numerik, yang memberikan kemampuan untuk bekerja dengan data tabular dan melakukan operasi matematika. Kemudian, kita mengatur tampilan output agar semua kolom dataset dapat ditampilkan dengan “pd.set_option”. Selanjutnya, kita membaca dataset dari file CSV dengan “pd.read_csv” dan melakukan beberapa operasi awal seperti menampilkan beberapa data pertama, mendapatkan informasi dataset, memeriksa nilai null, dan menghitung statistik deskriptif. Kode ini adalah langkah awal yang penting dalam analisis data, karena membantu kita memahami struktur dataset, mengidentifikasi masalah data seperti nilai null, dan mendapatkan gambaran awal tentang data yang akan dianalisis.

3. *Preprocessing Data*

```
# Import library
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
# Mengimpor StemmerFactory dari pustaka Sastrawi
import nltk
# Mengimpor pustaka nltk (Natural Language Toolkit)
import re
# Mengimpor modul re (regular expressions)

nltk.download('punkt')
# Mengunduh dataset 'punkt' dari nltk (digunakan untuk tokenisasi)
nltk.download('stopwords')
# Mengunduh dataset 'stopwords' dari nltk (digunakan untuk
menghapus kata-kata tidak penting)

# Define preprocessing function
def preprocess(text):
    # Remove punctuation
    text = re.sub(r'^\w\s', '', text)
    # Menghapus tanda baca menggunakan regular expressions
    # Lowercase
    text = text.lower()
    # Mengubah semua huruf menjadi huruf kecil
    # Tokenize
    text = nltk.word_tokenize(text)
    # Melakukan tokenisasi teks menjadi kata-kata
    # Remove stopwords
    stopwords = nltk.corpus.stopwords.words('indonesian')
    # Mengambil kata-kata tidak penting berbahasa Indonesia
    text = [word for word in text if word not in stopwords]
    # Menghapus kata-kata tidak penting
```

```

# Stemming
factory = StemmerFactory()
# Membuat objek factory dari StemmerFactory
stemmer = factory.create_stemmer()
# Membuat objek stemmer menggunakan factory
text = [stemmer.stem(word) for word in text]
# Melakukan stemming pada kata-kata
# Return preprocessed text
return text
# Mengembalikan teks yang telah diproses

# Function to cast a vector
def cast_vector(row):
    return np.array(list(map(lambda x: x.astype('double'), row)))
# Mengonversi elemen-elemen vektor menjadi tipe data 'double'

```

Kode di atas adalah implementasi proses preprocessing teks yang umumnya digunakan dalam analisis teks atau pemrosesan bahasa alami (NLP). Pertama, pustaka-pustaka yang diperlukan diimpor, seperti Sastrawi untuk stemming, nltk untuk tokenisasi dan penghapusan stopwords, dan modul re untuk menghapus tanda baca. Selanjutnya, dataset “punkt” dan “stopwords” diunduh untuk digunakan dalam proses tokenisasi dan penghapusan kata-kata tidak penting. Fungsi “preprocess(text)” melakukan langkah-langkah dasar preprocessing, termasuk penghapusan tanda baca, konversi ke huruf kecil, tokenisasi, penghapusan stopwords, dan stemming. Fungsi “cast_vector(row)” digunakan untuk mengonversi elemen-elemen vektor ke tipe data “double”. Kode ini dapat digunakan sebagai langkah awal dalam mempersiapkan teks mentah sebelum dilakukan analisis teks atau pemrosesan bahasa alami, yang dapat diterapkan dalam berbagai konteks bahasa dan analisis data.

4. *Training Data*

```

# Import library
import pandas as pd
# Mengimpor pustaka pandas untuk manipulasi data
import numpy as np
# Mengimpor pustaka numpy untuk operasi numerik
import gensim
# Mengimpor pustaka gensim untuk pemodelan bahasa alami
import joblib
# Mengimpor pustaka joblib untuk penyimpanan model
from sklearn.cluster import KMeans
# Mengimpor modul KMeans dari pustaka scikit-learn untuk clustering
from sklearn.metrics import accuracy_score
# Mengimpor metrik akurasi dari pustaka scikit-learn

# Load dataset
df = pd.read_csv('5000-Data.csv')
# Membaca dataset dari file CSV dengan nama '5000-Data.csv'

# Preprocessing data
# Remove punctuation, lowercase, tokenize, etc.

```



```

df['translated_description'] =
df['translated_description'].apply(preprocess)
# Memproses kolom 'translated_description' dengan fungsi
preprocess()

# Train Word2Vec model
model = gensim.models.Word2Vec(df['translated_description'],
vector_size=100, window=5, min_count=1, workers=4)
# Melatih model Word2Vec dengan parameter tertentu untuk melakukan
featured extraction dengan mengubah teks menjadi vektor numerik

# Get word vectors
word_vectors = model.wv
# Mendapatkan vektor kata dari model Word2Vec

# Save the model
model.save('word2vec.model')
# Menyimpan model Word2Vec ke dalam file 'word2vec.model'

# Get sentence vectors by averaging word vectors
sentence_vectors = []
for text in df['translated_description']:
    vector = np.mean([word_vectors[word] for word in text], axis=0)
    # Menghitung vektor kalimat dengan mengambil rata-rata vektor
    kata
    sentence_vectors.append(vector)

# Convert to numpy array
sentence_vectors = np.array(sentence_vectors).astype(np.double)
# Mengkonversi hasil vektor kalimat menjadi array numpy dengan
tipe data double

# Cluster data using K-Means
kmeans = KMeans(n_clusters=3, random_state=0)
# Menginisialisasi model K-Means dengan 3 cluster
kmeans.fit(sentence_vectors)
# Melakukan clustering pada vektor kalimat

# Save the model
joblib.dump(kmeans, 'kmeans.model')
# Menyimpan model K-Means ke dalam file 'kmeans.model'

# Get cluster labels
labels = kmeans.labels_
# Mendapatkan label cluster untuk setiap data

# Assign labels to data
df['label'] = labels
# Menambahkan kolom 'label' ke dataset yang berisi label cluster

```

Tabel di atas adalah implementasi langkah-langkah dalam analisis teks menggunakan pemodelan Word2Vec dan K-Means Clustering. Pertama, dataset dibaca menggunakan pustaka pandas dan kemudian diproses dengan langkah-langkah preprocessing, termasuk penghapusan tanda baca, konversi ke huruf kecil, tokenisasi, dan pengubahan teks ke dalam vektor numerik dengan model Word2Vec. Model Word2Vec dilatih dengan dataset untuk menghasilkan representasi vektor kata.

Selanjutnya, vektor kata tersebut digunakan untuk menghitung vektor kalimat dengan cara mengambil rata-rata vektor kata dalam setiap kalimat. Proses clustering dilakukan menggunakan K-Means Clustering dengan 3 cluster, dan label cluster diberikan pada setiap data. Hasil clustering ini dapat digunakan untuk mengelompokkan data teks menjadi kelompok-kelompok yang memiliki kesamaan dalam konteks kata yang digunakan. Model Word2Vec dan model K-Means Clustering kemudian disimpan untuk digunakan pada tahap analisis atau tugas lainnya. Keseluruhan proses ini bertujuan untuk memahami pola dan kelompok dalam data teks yang dapat membantu dalam pengambilan keputusan atau pemahaman konten teks yang besar dalam konteks yang lebih umum.

5. *Testing Model*

```
# Import library
import pandas as pd
# Mengimpor pustaka pandas untuk manipulasi data
import numpy as np
# Mengimpor pustaka numpy untuk operasi numerik
import gensim
# Mengimpor pustaka gensim untuk pemodelan bahasa alami
from sklearn.cluster import KMeans
# Mengimpor modul KMeans dari pustaka scikit-learn untuk clustering
from sklearn.metrics import accuracy_score
# Mengimpor metrik akurasi dari pustaka scikit-learn
import joblib
# Mengimpor pustaka joblib untuk penyimpanan model

# Load trained models
model = gensim.models.Word2Vec.load('word2vec.model')
# Memuat model Word2Vec yang telah dilatih dari file 'word2vec.model'
kmeans = joblib.load('kmeans.model')
# Memuat model K-Means yang telah dilatih dari file 'kmeans.model'

# Define mapping function
def map_label_to_sentiment(label):
    # You can use your own mapping function here
    if label == 0:
        return 'positive'
    elif label == 1:
        return 'negative'
    else:
        return 'neutral'

# Get text input from user
text = input('Masukkan Input: ')
# Mengambil input teks dari pengguna

# Preprocess text
# Remove punctuation, lowercase, tokenize, etc.
# You can use your own preprocessing function here
text = preprocess(text)
# Memproses teks input menggunakan fungsi preprocess
```

```

# Get word vectors
word_vectors = model.wv
# Mendapatkan vektor kata dari model Word2Vec

# Get sentence vector by averaging word vectors
sentence_vector = np.mean([word_vectors[word] for word in text],
axis=0)
# Menghitung vektor kalimat dengan mengambil rata-rata vektor kata

# Ubah tipe data menjadi double
sentence_vector = np.array(sentence_vector, dtype=np.double)
# Mengkonversi vektor kalimat menjadi tipe data double

# Prediksi label cluster
label = kmeans.predict([sentence_vector])[0]
# Memprediksi label cluster untuk vektor kalimat

# Map label to sentiment class
sentiment = map_label_to_sentiment(label)
# Mengonversi label cluster menjadi kelas sentimen

# Print sentiment result
print('Sentimen dari teks tersebut adalah:', sentiment)
# Mencetak hasil sentimen ke layar

```

Tabel di atas adalah implementasi yang memungkinkan pengguna untuk memasukkan teks tertentu dan kemudian memprediksi sentimen atau sikap yang mungkin terkandung dalam teks tersebut menggunakan model Word2Vec dan K-Means Clustering yang telah dilatih sebelumnya. Proses dimulai dengan memuat model Word2Vec dan model K-Means yang telah disimpan sebelumnya dari file. Pengguna kemudian diminta untuk memasukkan teks yang ingin dianalisis. Teks tersebut kemudian diolah, termasuk penghapusan tanda baca, konversi huruf menjadi huruf kecil, dan tokenisasi. Selanjutnya, vektor kalimat dibuat dengan mengambil rata-rata vektor kata yang terkandung dalam teks tersebut. Kemudian, dengan menggunakan model K-Means, label cluster diprediksi untuk vektor kalimat tersebut. Label cluster kemudian diubah menjadi kelas sentimen (positif, negatif, atau netral) dengan menggunakan fungsi pemetaan. Hasil sentimen dari teks tersebut kemudian dicetak ke layar.

3.2 SUMBER SOURCE CODE

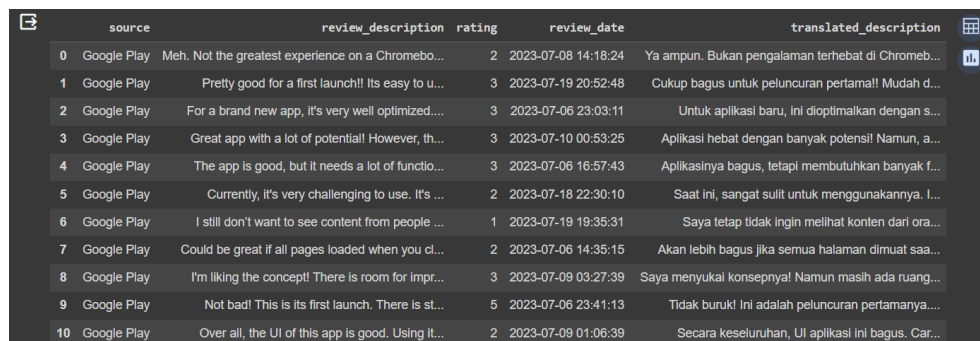
Sumber *source code* yang kami gunakan dalam penelitian ini dapat ditemukan di <https://www.kaggle.com/code/prashant111/k-means-clustering-with-python/notebook>. *Source code* tersebut memiliki peran krusial dalam pengembangan modul analisis sentimen untuk aplikasi Threads, dimana kami hanya mengadaptasi dan memodifikasinya sesuai kebutuhan kami dalam penelitian analisis sentimen aplikasi Threads. Sumber *source*

code tersebut digunakan sebagai referensi yang kami pelajari dan diubah sesuai dengan konteks dan tujuan penelitian ini.

Dengan memanfaatkan *source code* ini, kami dapat mengimplementasikan algoritma *K-Means Clustering* untuk mengelompokkan ulasan pengguna ke dalam kategori sentimen yang berbeda, yaitu “Positif”, “Negatif”, atau “Netral”. *Source code* ini menjadi landasan yang kuat dalam membangun modul analisis sentimen yang efisien dan akurat, yang memungkinkan kami untuk memahami dan mengevaluasi sentimen pengguna terhadap aplikasi Threads dengan lebih mendalam.

3.3 DATASET

Dataset yang digunakan dalam penelitian ini adalah dataset ulasan pengguna terhadap Aplikasi Threads yang bersumber dari Kaggle, dimana Anda dapat melihatnya pada tautan berikut <https://www.kaggle.com/datasets/saloni1712/threads-an-instagram-app-reviews>. Dataset ini awalnya berjumlah kurang lebih 32.910 data dalam bahasa Inggris. Namun, untuk mempermudah proses pelatihan model, penelitian ini melakukan penerjemahan terhadap seluruh 32.910 data tersebut dan memilih sebagian kecil, yaitu 5.000 data, untuk analisis lebih lanjut. Dataset ini berfokus pada ulasan pengguna terhadap aplikasi Threads, yang mencakup beragam pandangan dan opini terkait pengalaman penggunaan aplikasi tersebut.



	source	review_description	rating	review_date	translated_description
0	Google Play	Meh. Not the greatest experience on a Chromebo...	2	2023-07-08 14:18:24	Ya ampun. Bukan pengalaman terbaik di Chromebo...
1	Google Play	Pretty good for a first launch!! Its easy to u...	3	2023-07-19 20:52:48	Cukup bagus untuk peluncuran pertama!! Mudah d...
2	Google Play	For a brand new app, it's very well optimized...	3	2023-07-06 23:03:11	Untuk aplikasi baru, ini dioptimalkan dengan s...
3	Google Play	Great app with a lot of potential! However, th...	3	2023-07-10 00:53:25	Aplikasi hebat dengan banyak potensi! Namun, a...
4	Google Play	The app is good, but it needs a lot of functio...	3	2023-07-06 16:57:43	Aplikasinya bagus, tetapi membutuhkan banyak f...
5	Google Play	Currently, it's very challenging to use. It's ...	2	2023-07-18 22:30:10	Saat ini, sangat sulit untuk menggunakannya. I...
6	Google Play	I still don't want to see content from people ...	1	2023-07-19 19:35:31	Saya tetap tidak ingin melihat konten dari ora...
7	Google Play	Could be great if all pages loaded when you cl...	2	2023-07-06 14:35:15	Akan lebih bagus jika semua halaman dimuat saa...
8	Google Play	I'm liking the concept! There is room for impr...	3	2023-07-09 03:27:39	Saya menyukai konsepnya! Namun masih ada ruang...
9	Google Play	Not bad! This is its first launch. There is st...	5	2023-07-06 23:41:13	Tidak buruk! Ini adalah peluncuran pertamanya....
10	Google Play	Over all, the UI of this app is good. Using it...	2	2023-07-09 01:06:39	Secara keseluruhan, UI aplikasi ini bagus. Car...

Gambar 1.2 Preview 10 Data Teratas dari *Dataset*

Dataset ini terdiri dari beberapa atribut, diantaranya “source” untuk menunjukkan sumber datanya, “review_description” untuk menunjukkan ulasan asli dari pengguna aplikasi Threads, “rating” yang menunjukkan tingkat kepuasan pengguna dalam skala angka 1 sampai 5, “review_date” yang menunjukkan waktu ulasan dikirim pengguna, dan “translated_description” yang menunjukkan ulasan pengguna setelah diterjemahkan ke dalam Bahasa Indonesia.

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   source                 5000 non-null   object
1   review_description     5000 non-null   object
2   rating                 5000 non-null   int64
3   review_date            5000 non-null   object
4   translated_description 5000 non-null   object
dtypes: int64(1), object(4)
memory usage: 195.4+ KB

```

Gambar 1.3 Informasi Terkait *Dataset*

Selain itu, dalam proses pengolahan dataset yang digunakan dalam program ini, dilakukan pemeriksaan untuk setiap entri atau nilai dalam baris dataset guna mengidentifikasi apakah ada nilai yang kosong atau tidak terisi. Hal ini penting untuk memastikan kualitas data yang akan digunakan dalam analisis sentimen, karena nilai-nilai yang kosong dapat memengaruhi hasil analisis dan memerlukan penanganan khusus, seperti pengisian nilai yang hilang atau penghapusan data yang tidak lengkap. Pemeriksaan ini merupakan langkah awal dalam proses preprocessing data sebelum dilanjutkan ke tahap analisis sentimen.

BAB IV

HASIL DAN PEMBAHASAN

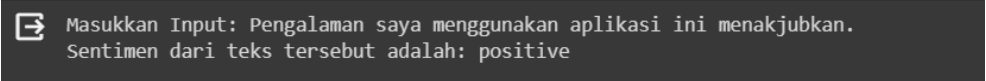
4.1 PENGUJIAN

Pengujian dilakukan dengan metode ekstraksi fitur yang digunakan dalam analisis sentimen menggunakan “Word2Vec”. Proses pelatihan “Word2Vec” melibatkan pemindaian teks ulasan yang ada dalam dataset untuk memahami hubungan antar kata dalam konteks tertentu. Setelah pelatihan selesai, akan dihasilkan vektor representasi numerik untuk setiap kata dalam kamus. Representasi ini dikenal sebagai *word embeddings* atau *word vectors*.

Kemudian, kami menggunakan metode *K-Means Clustering* untuk mengelompokkan ulasan-ulasan baru ini ke dalam tiga kelas sentimen yang telah ditentukan sebelumnya: positif, negatif, dan netral. Jumlah *cluster* ini dipilih berdasarkan hasil metode Elbow, yang mengindikasikan bahwa tiga *cluster* adalah jumlah yang optimal untuk analisis sentimen terhadap aplikasi Threads.

Pada analisis sentimen ini, kami juga melakukan pengujian ulasan baru terhadap aplikasi Threads, dimana pertama akan dilakukan *preprocessing* pada teks ulasan tersebut. *Preprocessing* mencakup langkah-langkah seperti penghapusan tanda baca, konversi teks menjadi huruf kecil, tokenisasi, penghapusan kata-kata tidak penting, dan *stemming*. Setelah *preprocessing* selesai, teks ulasan baru siap untuk diekstraksi fiturnya dengan menggunakan model “Word2Vec” yang telah dilatih sebelumnya.

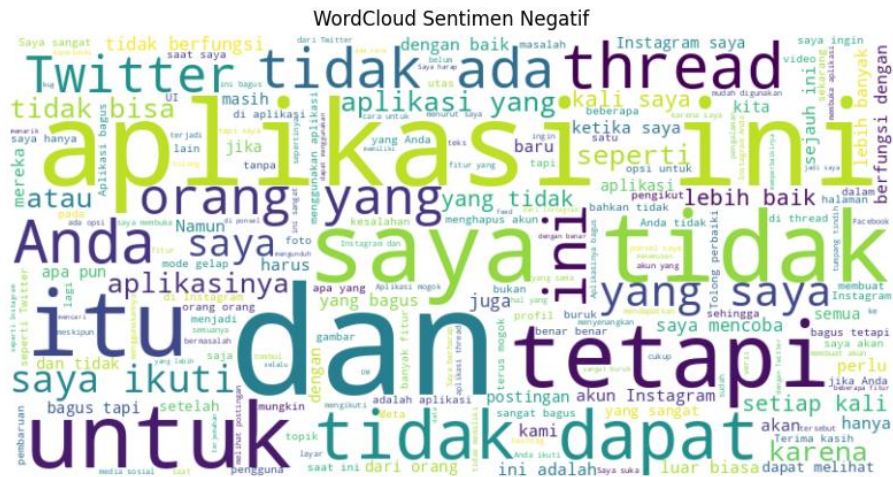
Setelah teks ulasan baru diubah menjadi representasi vektor numerik dengan model “Word2Vec”, langkah selanjutnya adalah melakukan *clustering* menggunakan algoritma *K-Means*. Dalam kasus ini, kami telah menentukan sejumlah 3 *cluster* (positif, negatif, dan netral). Setiap ulasan baru akan di-*assign* ke salah satu *cluster* berdasarkan kemiripan vektornya dengan *centroid* dari masing-masing *cluster*.



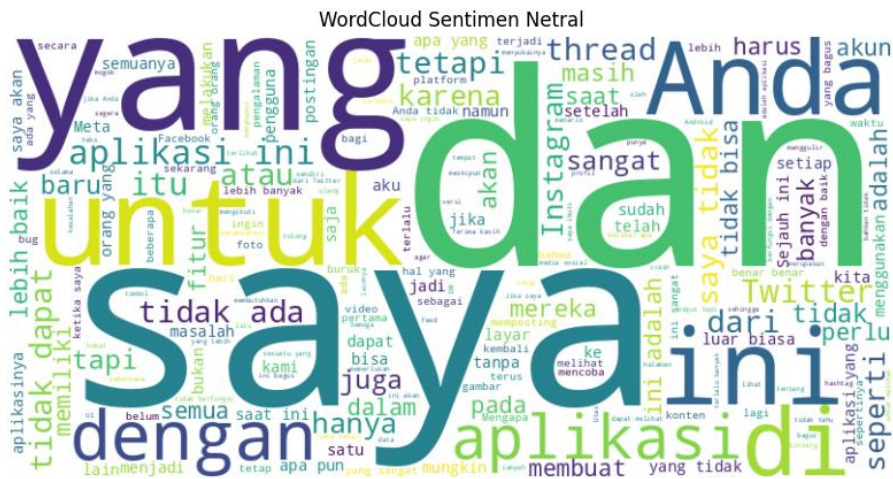
```
Masukkan Input: Pengalaman saya menggunakan aplikasi ini menakjubkan.  
Sentimen dari teks tersebut adalah: positive
```

Gambar 1.4 Pengujian Data Baru dengan Ulasan Positif

Berdasarkan gambar di atas, terlihat bahwa program telah menganalisis ulasan pengguna terhadap aplikasi Threads dan menghasilkan sentimen “Positive” untuk ulasan tersebut. Ulasan pengguna menyatakan bahwa “Pengalaman saya menggunakan aplikasi ini menakjubkan.” Ini menunjukkan bahwa pengguna memberikan ulasan yang sangat



Gambar 1.8 *Word Cloud* Sentimen Negatif



Gambar 1.9 *Word Cloud* Sentimen Netral

Melalui gambar-gambar *Word Cloud* yang telah disajikan, terlihat bahwa program membantu dalam mengidentifikasi kata-kata kunci yang sering muncul dalam ulasan positif, negatif, dan netral. *Word Cloud* ini memberikan wawasan visual yang kuat tentang kata-kata yang mencirikan sentimen pengguna terhadap aplikasi Threads. Namun, perlu diingat bahwa terkadang beberapa kata-kata yang muncul dalam *Word Cloud* tidak selalu terdistribusi dengan tepat ke dalam sentimen tertentu. Artinya, ada kata-kata tertentu yang mungkin sering digunakan dalam berbagai konteks ulasan, sehingga dapat muncul dalam *Word Cloud* untuk berbagai sentimen.

4.2 PEMBAHASAN

Dalam bagian ini, kami akan mengevaluasi hasil analisis sentimen yang ditemukan, meskipun tanpa menggunakan label manual yang konkret. Karena analisis ini bergantung

pada teknik *machine learning* untuk menganalisis sentimen, kami akan membahas pendekatan yang digunakan dan wawasan yang ditemukan.

1. Pendekatan Evaluasi

Dalam analisis sentimen kami, kami menggunakan pendekatan berbasis *K-Means Clustering*. Kami tidak memiliki label manual yang konkret untuk menentukan sentimen, sehingga evaluasi lebih bersifat eksploratif. Kami mengelompokkan ulasan-ulasan yang serupa berdasarkan representasi vektor mereka dalam ruang fitur Word2Vec.

2. Metrik Evaluasi

Meskipun tanpa label manual, kita dapat menggunakan beberapa metrik internal untuk mengukur kualitas *clustering*, seperti inersia (*inertia*) yang mengukur sejauh mana titik-titik dalam satu *cluster* mendekati pusat *cluster*-nya. Meskipun metrik ini memberikan gambaran tentang bagaimana titik-titik dalam satu *cluster* terkonsentrasi, ini bukan metrik yang eksplisit mengukur sentimen. Oleh karena itu, perlu dicatat bahwa hasil evaluasi kami lebih pada struktur data dan kemiripan ulasan daripada sentimen sebenarnya.

3. Wawasan yang Ditemukan

Hasil analisis kami mengungkapkan adanya sejumlah *cluster* dengan karakteristik tertentu yang terdiri atas sebagai berikut.

- a. Klaster Sentimen Positif → Klaster ini berisi ulasan-ulasan yang cenderung memiliki sentimen positif terhadap aplikasi Threads. Pengguna dalam klaster ini mungkin mengungkapkan kepuasan atau kegembiraan dengan aplikasi tersebut.
- b. Klaster Sentimen Negatif → Klaster ini berisi ulasan-ulasan yang cenderung memiliki sentimen negatif terhadap aplikasi Threads. Pengguna dalam klaster ini mungkin memiliki keluhan atau ketidakpuasan terhadap beberapa aspek aplikasi tersebut.
- c. Klaster Sentimen Netral → Klaster ini berisi ulasan-ulasan yang cenderung memiliki sentimen netral atau tidak jelas. Pengguna dalam klaster ini mungkin memberikan ulasan yang tidak mengandung evaluasi sentimen yang kuat.

4. Keterbatasan Analisis

Penting untuk mencatat bahwa karena analisis ini tidak bergantung pada label manual, ada keterbatasan dalam menentukan dengan pasti sentimen sebenarnya dari setiap ulasan. Hasilnya lebih pada pengelompokkan ulasan berdasarkan kemiripan dalam representasi vektor kata. Sebagai hasilnya, beberapa ulasan mungkin

dikelompokkan secara keliru, dan interpretasi sentimen perlu dilakukan dengan hati-hati.

Meskipun ada keterbatasan dalam evaluasi sentimen tanpa label manual yang konkret, analisis ini dapat memberikan pemahaman awal tentang bagaimana pengguna merasakan aplikasi Threads. Pengembang dapat menggunakan wawasan ini sebagai panduan awal untuk membuat perbaikan atau perubahan yang sesuai pada aplikasi mereka.

BAB V

PENUTUP

5.1 KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, kami dapat menyimpulkan beberapa poin utama sebagai berikut.

1. Kesimpulan Sentimen

Hasil analisis kami mengindikasikan bahwa terdapat tiga klaster utama dalam ulasan pengguna aplikasi Threads yang terdiri atas sebagai berikut.

- a. Klaster Sentimen Positif: Berisi ulasan-ulasan yang cenderung memiliki sentimen positif terhadap aplikasi Threads.
- b. Klaster Sentimen Negatif: Berisi ulasan-ulasan yang cenderung memiliki sentimen negatif terhadap aplikasi Threads.
- c. Klaster Sentimen Netral: Berisi ulasan-ulasan yang cenderung memiliki sentimen netral atau tidak jelas.

2. Keterbatasan Sentimen

Kami ingin mengingatkan bahwa hasil analisis sentimen kami tidak bersifat absolut dan bergantung pada pengelompokan berdasarkan kemiripan dalam representasi vektor kata-kata. Keterbatasan utama adalah kurangnya label sentimen manual yang konkret untuk validasi.

5.2 SARAN

Berdasarkan kesimpulan yang telah didapatkan, kami merekomendasikan beberapa langkah yang dapat diambil untuk meningkatkan aplikasi Threads dan meningkatkan kepuasan pengguna.

1. Analisis Lebih Lanjut

Sebagai langkah awal, kami menyarankan melakukan analisis lebih lanjut untuk mendapatkan pemahaman yang lebih mendalam tentang ulasan-ulasan dalam masing-masing *cluster*. Ini dapat melibatkan analisis teks secara manual untuk memahami permasalahan yang mungkin dihadapi pengguna dalam *cluster* sentimen negatif.

2. Perbaikan Produk

Berdasarkan analisis sentimen, pengembang dapat mengidentifikasi area-area tertentu yang memicu sentimen negatif. Ini dapat digunakan sebagai panduan untuk

perbaikan produk, termasuk perbaikan fitur atau pengurangan masalah yang disoroti oleh pengguna.

3. Komunikasi dengan Pengguna

Mengambil umpan balik dari pengguna, terutama dari *cluster* sentimen negatif, dan berkomunikasi secara aktif dengan mereka untuk memahami masalah mereka lebih dalam dan menawarkan solusi yang sesuai.

4. Pelatihan Algoritma Sentimen

Jika memungkinkan, pengembang dapat mempertimbangkan melatih model sentimen berdasarkan label manual untuk meningkatkan akurasi analisis sentimen di masa mendatang.

5. Monitoring Terus-Menerus

Penting untuk terus memonitor ulasan pengguna dan memperbarui analisis sentimen secara berkala untuk mengidentifikasi perubahan tren dan mengambil tindakan yang diperlukan.

DAFTAR PUSTAKA

- Budi, S. (2017). Text mining untuk analisis sentimen review film menggunakan algoritma K-Means. *Techno. Com*, 16(1), 1-8.
- Ediyanto, M. N. M., & Satyahadewi, N. (2013). Pengklasifikasian Karakteristik Dengan Metode K-Means Cluster Analysis. *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, 2(02).
- Giovani, A. P., Ardiansyah, A., Haryanti, T., Kurniawati, L., & Gata, W. (2020). Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi. *Jurnal Teknoinfo*, 14(2), 115-123.
- Syaifudin, Y. W., & Irawan, R. A. (2018). Implementasi Analisis Clustering Dan Sentimen Data Twitter Pada Opini Wisata Pantai Menggunakan Metode K-Means. *Jurnal Informatika Polinema*, 4(3), 189-194.