

IMPLEMENTASI DATA MINING DALAM MENGELOMPOKKAN PENGGUNA APLIKASI THREADS MENGGUNAKAN K-MEANS CLUSTERING

Linda Novia Rizky Lubis

lindalubis492@gmail.com

Universitas Islam Negeri Sumatera Utara

Informasi Artikel

Diterima : 01-12-2025

Direview : 03-12-2025

Disetujui : 20-12-2025

Abstrak

Threads media sosial baru besutan Meta sempat mencatat lonjakan pengguna secara signifikan setelah peluncuran, namun mengalami penurunan aktivitas dalam waktu singkat. Penelitian ini bertujuan mengelompokkan pengguna aplikasi Threads berdasarkan frekuensi penggunaan (menit/minggu) dan faktor penyebab penurunan minat menggunakan algoritma K-Means Clustering. Data diperoleh melalui survei daring terhadap 117 mahasiswa angkatan 2020. Metode Elbow digunakan untuk menentukan jumlah cluster optimal, yang menghasilkan tiga kelompok pengguna, yaitu pengguna jarang, pengguna sedang, dan pengguna aktif. Hasil evaluasi menggunakan Davies-Bouldin Index (DBI) menunjukkan nilai 0,5195 yang menandakan kualitas cluster cukup baik. Faktor utama penurunan minat meliputi kurangnya interaksi dengan teman, desain antarmuka yang kurang familiar, serta kekhawatiran terhadap keamanan data.

Keywords

Data Mining, K-Means Clustering, Threads, Elbow Method, Davies-Bouldin Index

Abstrak

Meta's new social media platform Threads saw a significant increase in users after its launch, but experienced a short-term decline in activity. This study aims to group Threads application users based on weekly usage frequency (minutes/week) and factors contributing to declining interest using the K-Means Clustering algorithm. Data were obtained through an online survey of 117 students from the 2020 cohort. The Elbow Method was used to determine the optimal number of clusters, resulting in three user groups: infrequent, moderate, and active users. Evaluation using the Davies-Bouldin Index (DBI) yielded a value of 0.5195, indicating fairly good cluster quality. The main factors contributing to decreased user interest include limited interaction with friends, unfamiliar interface design, and concerns about data security.

A. Pendahuluan

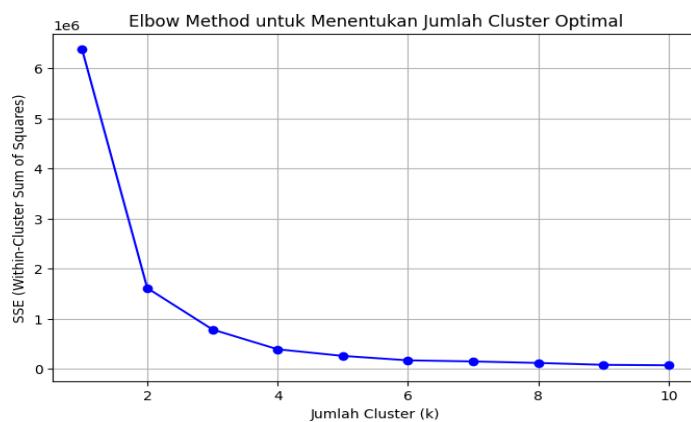
Media sosial saat ini telah menjadi elemen yang tidak dapat dipisahkan dari kehidupan masyarakat modern. Di Indonesia sendiri, jumlah pengguna media sosial diperkirakan telah mencapai sekitar 139 juta orang [1]. Salah satu platform terbaru yang menarik perhatian adalah *threads*, aplikasi media sosial besutan yang berhasil memperoleh lebih dari 190 juta pengguna di seluruh dunia. Meskipun pertumbuhannya sangat pesat, *threads* menghadapi tantangan dalam mempertahankan tingkat keterlibatan penggunanya. Diluncurkan oleh Meta pada tahun 2023 sebagai respon terhadap perubahan *lanskap microblogging* pasca-akuisisi Twitter oleh Elon Musk [2]. *Threads* mengalami pertumbuhan yang sangat cepat pada awal peluncurannya. Dalam lima hari pertama, *Threads* berhasil mencapai 100 juta pendaftaran, menjadikannya aplikasi dengan pertumbuhan tercepat dalam sejarah. Hingga awal tahun 2024, *platform* ini telah memiliki sekitar 160 juta pengguna aktif bulanan, dan pada Agustus 2024, jumlah tersebut meningkat hingga 190 juta pengguna global [3]. Namun kemudian mengalami penurunan aktivitas secara signifikan. Kondisi ini menandakan perlunya analisis untuk memahami pola penggunaan dan faktor penyebab penurunan tersebut. Data mining, khususnya algoritma K-Means Clustering, dapat dimanfaatkan untuk mengelompokkan pengguna berdasarkan perilaku penggunaan. Penelitian ini berfokus untuk mengidentifikasi karakteristik pengguna Threads melalui pendekatan clustering serta memberikan gambaran kelompok pengguna yang memiliki potensi untuk terus aktif ataupun berpotensi meninggalkan aplikasi [4].

B. Metode Penelitian

Penelitian ini menggunakan metode K-Means Clustering dalam mengelompokkan pengguna Threads berdasarkan dua variabel, yaitu frekuensi penggunaan (menit/minggu) dan faktor penyebab penurunan minat [5]. Data dikumpulkan melalui survei daring terhadap 117 mahasiswa angkatan 2020. Tahapan penelitian meliputi pengumpulan data, preprocessing data (termasuk encoding kategori), penentuan jumlah cluster optimal menggunakan Elbow Method, implementasi algoritma K-Means, dan evaluasi hasil cluster menggunakan Davies-Bouldin Index (DBI) [6].

C. Hasil dan Pembahasan

Hasil analisis Elbow Method menunjukkan bahwa tiga cluster merupakan jumlah optimal untuk data tersebut [7]. Cluster yang terbentuk adalah: (1) pengguna jarang, (2) pengguna sedang, dan (3) pengguna aktif. Menentukan jumlah *cluster* dan nilai *centroid* awal [8]. Setelah menggambarkan data awal, langkah berikutnya adalah menentukan jumlah *cluster* yang paling sesuai. Dalam penelitian ini, metode *Elbow* digunakan untuk menemukan jumlah *cluster* terbaik yang akan diterapkan dalam algoritma *K-Means Clustering*. Berdasarkan hasil perhitungan, diperoleh grafik *Elbow Method* sebagai berikut:

**Gambar 1. Grafik Elbow Method**

Dari grafik tersebut, tampak bahwa penurunan WCSS mulai melambat saat jumlah *cluster* mencapai tiga. Karenanya, jumlah *cluster* yang paling tepat digunakan dalam penelitian ini adalah tiga *cluster*, yaitu:

1. *Cluster 0 (C0)* - Pengguna Jarang
2. *Cluster 1 (C1)* - Pengguna Sedang
3. *Cluster 2 (C2)* - Pengguna Aktif

Adapun nilai *centroid* data awal diperoleh dari nilai terendah, nilai rata-rata (*mean*), dan nilai tertinggi pada Tabel 4.1 yaitu:

$$C_0 = 72$$

$$C_1 = 390$$

$$C_2 = 882$$

Menghitung jarak ke *centroid* pertama kali & menentukan *cluster* awal. Berikut adalah perhitungan jarak setiap data terhadap C0, perhitungan ini didasarkan pada jarak terdekat dari setiap pengguna dengan *centroid* awal menggunakan Persamaan (2.1) :

$$\text{Pengguna 1} = \sqrt{(390 - 72)^2} = 318$$

$$\text{Pengguna 2} = \sqrt{(156 - 72)^2} = 84$$

$$\text{Pengguna 3} = \sqrt{(198 - 72)^2} = 126$$

$$\text{Pengguna 4} = \sqrt{(78 - 72)^2} = 6$$

$$\text{Pengguna 5} = \sqrt{(192 - 72)^2} = 120$$

$$\text{Pengguna 6} = \sqrt{(276 - 72)^2} = 204$$

$$\text{Pengguna 7} = \sqrt{(570 - 72)^2} = 498$$

$$\text{Pengguna 8} = \sqrt{(384 - 72)^2} = 312$$

$$\text{Pengguna 9} = \sqrt{(672 - 72)^2} = 600$$

$$\text{Pengguna 10} = \sqrt{(444 - 72)^2} = 372$$

$$\text{Pengguna 117} = \sqrt{(774 - 72)^2} = 702$$

Berikut merupakan perhitungan jarak pada setiap data pada C1 menggunakan

$$\sqrt{(390 - 390)^2}$$

$$\sqrt{(156 - 390)^2}$$

Persamaan (2.1) :

$$\begin{aligned}
 \text{Pengguna 1} &= 0 \\
 \text{Pengguna 2} &= 234 \\
 \text{Pengguna } 3 &= \sqrt{(198 - 390)^2} = 192 \\
 \text{Pengguna } 4 &= \sqrt{(78 - 390)^2} = 312 \\
 \text{Pengguna } 5 &= \sqrt{(192 - 390)^2} = 198 \\
 \text{Pengguna } 6 &= \sqrt{(276 - 390)^2} = 114 \\
 \text{Pengguna } 7 &= \sqrt{(570 - 390)^2} = 180 \\
 \text{Pengguna } 8 &= \sqrt{(384 - 390)^2} = 6 \\
 \text{Pengguna } 9 &= \sqrt{(672 - 390)^2} = 282 \\
 \text{Pengguna } 10 &= \sqrt{(444 - 390)^2} = 54 \\
 \text{Pengguna } 11 &= \sqrt{(774 - 390)^2} = 384
 \end{aligned}$$

Berikut merupakan perhitungan jarak pada setiap data pada C1 menggunakan Persamaan (2.1) :

$$\begin{aligned}
 \text{Pengguna } 1 &= \sqrt{(390 - 882)^2} = 492 \\
 \text{Pengguna } 2 &= \sqrt{(156 - 882)^2} = 726 \\
 \text{Pengguna } 3 &= \sqrt{(198 - 882)^2} = 684 \\
 \text{Pengguna } 4 &= \sqrt{(78 - 882)^2} = 804 \\
 \text{Pengguna } 5 &= \sqrt{(192 - 882)^2} = 690 \\
 \text{Pengguna } 6 &= \sqrt{(276 - 882)^2} = 606 \\
 \text{Pengguna } 7 &= \sqrt{(570 - 882)^2} = 312 \\
 \text{Pengguna } 8 &= \sqrt{(384 - 882)^2} = 498 \\
 \text{Pengguna } 9 &= \sqrt{(672 - 882)^2} = 210 \\
 \text{Pengguna } 10 &= \sqrt{(444 - 882)^2} = 438 \\
 \text{Pengguna } 11 &= \sqrt{(774 - 882)^2} = 108
 \end{aligned}$$

Menentukan *cluster* awal dan menghitung kembali nilai *centroid*. Jika nilai terkecil ada di *cluster* 0 (C0), maka masuk ke *cluster* C0 dan seterusnya. Hasil dari proses pembagian kelompok pada iterasi pertama bisa dilihat di Tabel 1

Tabel 1 Hasil Pengelompokkan Iterasi 1

Cluster	Nilai
C0	42
C1	43
C2	32

Tentukan ulang titik tengah dari *cluster* baru berdasarkan rata-rata nilai dalam *cluster* dibagi jumlah data yang ada di *cluster* tersebut.

$$C0 = 207.75$$

$$C1 = 406.23$$

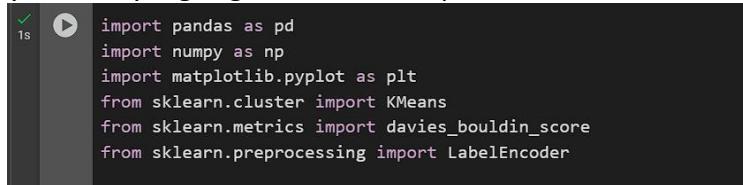
$$C2 = 677.62$$

Lakukan kembali langkah 3 hingga titik pusat setiap *cluster* tidak berubah. sistem akan menjalankan kembali perhitungan menggunakan nilai titik pusat *Cluster (Centroid)* terbaru [8]. Hingga kondisi untuk menghentikan proses *K-Means Clustering* terpenuhi, yaitu ketika hasil pengelompokan dan nilai *centroid* tidak berubah dari iterasi sebelumnya, proses ini akan terus berlangsung. Karena pada iterasi ke-2 tidak ada perubahan nilai *centroid*, maka hasil akhir yang diperoleh adalah $C0 = 42$, $C1 = 43$, dan $C2 = 32$. Penilaian kualitas cluster menunjukkan nilai DBI 0,5195 yang mengindikasikan bahwa cluster cukup terpisah dan memiliki koherensi internal yang baik. Faktor utama penurunan penggunaan Thread mencakup kurangnya interaksi, ketidaknyamanan antarmuka, serta kekhawatiran mengenai keamanan data. Temuan ini dapat digunakan pihak pengembang untuk meningkatkan fitur serta strategi retensi pengguna [9].

Penerapan K-Means *Clustering* pada Python

Dalam proses pengujian penerapan algoritma K-Means *Clustering*, penulis menggunakan bahasa pemrograman *Python* yang dijalankan melalui *platform Google Colaboratory* [10]. *Platform* ini dipilih karena mendukung lingkungan pengembangan berbasis *cloud*, integrasi dengan *Google Drive*, serta kemudahan dalam menjalankan berbagai *library* data *science* tanpa perlu instalasi lokal [1]].

Beberapa *library* utama yang digunakan dalam penelitian ini antara lain:



```

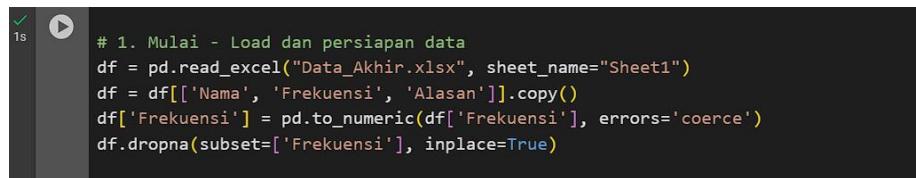
1s  ⏎
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
from sklearn.preprocessing import LabelEncoder

```

Gambar 2. Import Library

1. *pandas* untuk memproses dan memanipulasi data termasuk membersihkan, mengubah, menggabungkan, dan menganalisis data secara efisien
2. *numpy* untuk perhitungan numerik,
3. *matplotlib.pyplot* untuk membuat berbagai jenis visualisasi data.
4. K-Means dari *sklearn.cluster* untuk proses *clustering*,
5. *davies_bouldin_score* dari *sklearn.metrics* untuk evaluasi hasil *clustering*,
6. *LabelEncoder* dari *sklearn.preprocessing* untuk mengubah data kategorikal menjadi numerik.

Library-library di atas sudah tersedia secara *default* di Google Colab, sehingga mempermudah proses implementasi dan analisis tanpa memerlukan konfigurasi tambahan [12].



```
# 1. Mulai - Load dan persiapan data
df = pd.read_excel("Data_Aakhir.xlsx", sheet_name="Sheet1")
df = df[['Nama', 'Frekuensi', 'Alasan']].copy()
df['Frekuensi'] = pd.to_numeric(df['Frekuensi'], errors='coerce')
df.dropna(subset=['Frekuensi'], inplace=True)
```

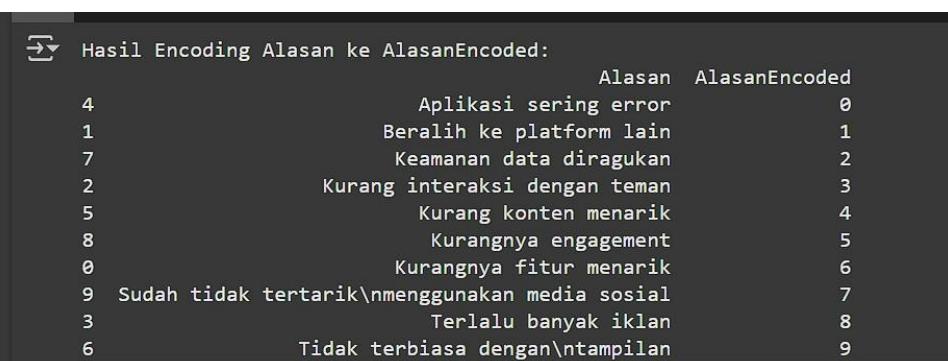
Gambar 3. Persiapan Data

Data dibaca dari *file Excel* dan hanya kolom-kolom yang relevan (Nama, Frekuensi, Alasan) yang digunakan [13]. Selanjutnya, dilakukan pembersihan data dengan memastikan bahwa nilai pada kolom Frekuensi merupakan numerik, serta menghapus data yang memiliki nilai kosong. Karena algoritma K-Means hanya dapat memproses data numerik, maka kolom Alasan yang bersifat kategorikal perlu diubah menjadi angka menggunakan *LabelEncoder*.



```
# 2. Encode alasan ke angka untuk clustering dan visualisasi
le = LabelEncoder()
df['AlasanEncoded'] = le.fit_transform(df['Alasan'])

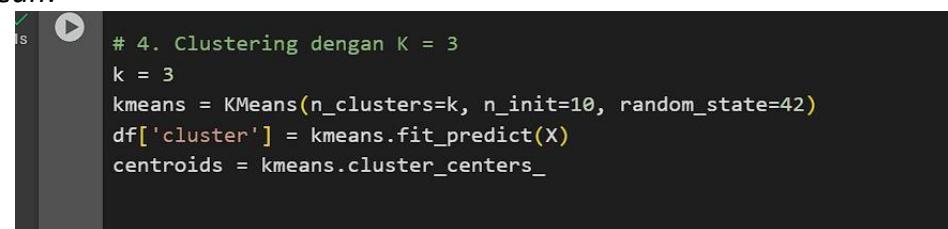
# Menampilkan hasil encoding
print("Hasil Encoding Alasan ke AlasanEncoded:")
print(df[['Alasan', 'AlasanEncoded']].drop_duplicates().sort_values(by='AlasanEncoded'))
```

Gambar 4. Proses Encoding


		Alasan	AlasanEncoded
4	Aplikasi sering error		0
1	Beralih ke platform lain		1
7	Keamanan data diragukan		2
2	Kurang interaksi dengan teman		3
5	Kurang konten menarik		4
8	Kurangnya engagement		5
0	Kurangnya fitur menarik		6
9	Sudah tidak tertarik\nmenggunakan media sosial		7
3	Terlalu banyak iklan		8
6	Tidak terbiasa dengan\ntampilan		9

Gambar 5. Output Encoding

Pengelompokan data menggunakan algoritma *K-Means Clustering*. Algoritma ini bekerja dengan cara memilih tiga *centroid* awal, kemudian mengelompokkan setiap data ke *centroid* yang paling dekat berdasarkan jarak *Euclidean*.



```
# 4. Clustering dengan K = 3
k = 3
kmeans = KMeans(n_clusters=k, n_init=10, random_state=42)
df['cluster'] = kmeans.fit_predict(X)
centroids = kmeans.cluster_centers_
```

Gambar 6. Implementasi K-Means Clustering

Untuk memudahkan interpretasi, setiap *cluster* diberi label berdasarkan rata-rata Frekuensi dari masing-masing *cluster*. *Cluster* dengan rata-rata frekuensi rendah diberi label "Pengguna Jarang", sedang untuk nilai sedang dan tinggi

diberi label yang sesuai.

```
# 5. Mapping cluster ke label bermakna
cluster_mean = df.groupby('cluster')['Frekuensi'].mean().sort_values()
cluster_labels = {
    cluster_mean.index[0]: 'Pengguna Jarang',
    cluster_mean.index[1]: 'Pengguna Sedang',
    cluster_mean.index[2]: 'Pengguna Aktif'
}
df['cluster_label'] = df['cluster'].map(cluster_labels)
```

Gambar 7. Pemberian Label pada *Cluster*

Untuk mengetahui distribusi data pada masing-masing *cluster*, ditampilkan jumlah anggota dari setiap *cluster*.

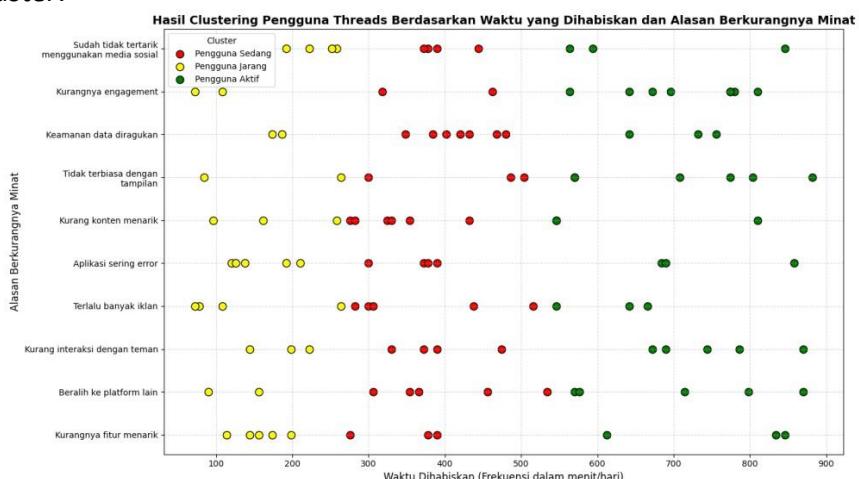
```
# 6. Menampilkan jumlah anggota tiap cluster
print("\nJumlah anggota setiap cluster:")
print(df['cluster_label'].value_counts())
```

Gambar 8. Menampilkan Jumlah Anggota Tiap *Cluster*

```
Jumlah anggota setiap cluster:
cluster_label
Pengguna Sedang      43
Pengguna Jarang      42
Pengguna Aktif       32
Name: count, dtype: int64
```

Gambar 9. Jumlah Tiap *Cluster*

Data divisualisasikan dalam bentuk *scatter plot* berdasarkan Frekuensi dan Alasan *Encoded*. Warna dan label digunakan untuk membedakan masing-masing *cluster*.



Gambar 10. *Scatter Plot*

4.2.1 Evaluasi *Davies-Bouldin Index* (DBI)

Setelah proses pengelompokan dilakukan dengan algoritma *K-Means*

Clustering, kualitas hasilnya dinilai menggunakan metode *Davies-Bouldin Index* (DBI). Tujuan dari evaluasi ini adalah untuk mengetahui seberapa baik *cluster* yang terbentuk memiliki bentuk yang padat dan terpisah, dua hal yang penting dalam pengelompokan data. Berikut adalah langkah – langkah menghitung nilai DBI:

1. *Centroid* Tiap *Cluster* (Rata-rata Frekuensi) C0 (Pengguna Jarang):

194.71

C1 (Pengguna

Sedang): 446.37 C2

(Pengguna Aktif):

752.44

2. Rata-rata Penyebaran

Data (S_i) S_o (C0) =

66.56

S_1 (C1)

=

71.86

S_2 (C2)

=

67.54

3. Jarak Antar

Centroid (M_{ij})

$M(C_0, C_1) =$

251.66

$M(C_0, C_2)$

= 557.72

$M(C_1, C_2)$

= 306.07

4. Nilai Maksimum R_{ij} Tiap

Cluster R_o (C0) =

$\max(R_{o1}, R_{o2}) = 0.5500 R_1$

(C1) = $\max(R_{10}, R_{12}) =$

0.5535 R_2 (C2) =

$\max(R_{20}, R_{21}) = 0.4552$

5. *Davies-Bouldin*

Index (DBI) DBI =

$$\frac{1}{k} \sum_{i=1}^k \max_{j \neq i} R_{ij}$$

$$\text{DBI} = \frac{1}{3} (0.5500 + 0.5535 + 0.4552) = 0,5195$$

```
# 8. Evaluasi dengan Davies-Bouldin Index
dbi = davies_bouldin_score(X, df['cluster'])
print(f"Davies-Bouldin Index (DBI): {dbi:.4f}")

Davies-Bouldin Index (DBI): 0.5195
```

Gambar 11. Nilai DBI

Hasil dari perhitungan pada *dataset* pengguna *Threads* menunjukkan bahwa nilai *Davies-Bouldin Index* (DBI) adalah: 0.5195. Nilai ini berada dalam rentang baik (<1), yang berarti bahwa setiap *cluster* yang terbentuk cenderung kompak (anggota berada dekat dengan *centroid*). Antar *cluster* juga memiliki pemisahan yang cukup baik, sehingga meminimalkan tumpang tindih. Pengelompokan ke dalam tiga *cluster* (pengguna jarang, sedang, dan aktif) berhasil memisahkan karakteristik pengguna dengan baik berdasarkan data frekuensi dan alasan berkurangnya minat terhadap aplikasi *Threads*.

D. Simpulan

Penelitian ini berhasil menerapkan K-Means Clustering untuk mengelompokkan pengguna *Threads* ke dalam tiga kategori berdasarkan frekuensi penggunaan dan penyebab penurunan minat. Hasil clustering menunjukkan kualitas pengelompokan yang cukup baik dengan nilai DBI 0,5195. Penelitian ini memberikan gambaran mengenai perilaku pengguna serta faktor yang mendorong menurunnya minat, yang dapat dijadikan acuan pengembangan fitur dan strategi peningkatan engagement.

E. Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada semua pihak yang berkontribusi dalam proses penelitian ini, termasuk para responden dan pihak Universitas Islam Negeri Sumatera Utara.

F. Referensi

- [1] Adani, N. F., Boy, A. F., & Syahputra, R. (2019). Implementasi Data Mining Untuk Pengelompokan Data Penjualan Menggunakan Algoritma K-Means. *Jurnal CyberTech*.
- [2] Amalina, T., Pramana, D. B. A., & Sari, B. N. (2022). Metode K-Means Clustering Dalam Pengelompokan Penjualan Produk Frozen Food. *Jurnal Ilmiah Wahana Pendidikan*.
- [3] Afiasari, N., Suarna, N., & Rahaningsih, N. (2023). Implementasi Data Mining Transaksi Penjualan Menggunakan K-Means.
- [4] Ariati, I., Norsa, R. N., Akhsan, L., & Heikal, J. (2023). Segmentasi Pelanggan Menggunakan K-Means Clustering.
- [5] Setya Hadi, H. (2025). INTERNET OF THING (IOT): PRINSIP DAN IMPLEMENTASINYA.

- [6] Yudhistira, A., & Andika, R. (2023). Pengelompokan Nilai Siswa Menggunakan K-Means.
- [7] Abdurrohman, M. H., et al. (2024). Implementasi K-Means pada Data Mahasiswa Baru.
- [8] S. Hadi and U. Nehe, "Digitalisasi Manajemen Inventory ATK Berbasis Web pada Balai Diklat Industri Padang ", JENTIK, vol. 3, no. 1, pp. 7-25, Apr. 2025.
- [9] Zaki, A., Irwan, & Sembe, I. A. (2022). Penerapan K-Means dalam Pengelompokan Profil Mahasiswa.
- [10] Sembiring, S. S. B., Nasyuha, A. H., & Mariami, I. (2020). Pengelompokan Data Siswa untuk Beasiswa Menggunakan K-Means.
- [11] Astuti, H. (2020). Penerapan K-Means Clustering untuk Pelanggan PT. Pinus Merah Abadi.
- [12] Meta & Kepios Reports (2024). Data Perkembangan Threads Tahun 2024.
- [13] M. Azizah and H. S. Hadi, "PERANCANGAN SISTEM ANTRIAN DENGAN SPEECH RECOGNITION BERBASIS WEB PADA PUSKESMAS GASAN GADANG KAB. PADANG PARIAMAN", JENTIK, vol. 2, no. 3, pp. 154-160, Dec. 2024.