Komputika: Jurnal Sistem Komputer
Volume 14, Nomor 1, April 2025, hlm. 41-51
DOI: 10.34010/hpm5tg52

p-ISSN: 2252-9039
e-ISSN: 2655-3198

Parallel Computing pada Clustering K-Means untuk Analisis Keketatan Program Studi SNBT 2023

Alif Faturahman Firdaus[1*], Azzahra Fahriza Fitriani [2], Eddy Prasetyo Nugroho[3]

[1, 2, 3)] Program Studi Ilmu Komputer, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia
Jl. Dr. Setiabudhi No. 229, Bandung, Indonesia 40154
email: [1] aliffaturahman@upi.edu, [2] azzahrafahriiza@upi.edu, [3] eddypn@upi.edu

(Naskah masuk: 28 Desember 2024; direvisi: 5 Februari 2025; diterima untuk diterbitkan: *30 April 2025*)

ABSTRAK *– Penelitian ini bertujuan untuk menganalisis keketatan program studi pada data SNBT tahun 2023 menggunakan metode Knowledge Discovery in Databases (KDD) dan algoritma K-Means Clustering. Keketatan program studi diukur berdasarkan rasio antara jumlah pendaftar dan daya tampung, mencerminkan tingkat persaingan dan popularitas program studi. Terdapat dua permasalahan utama yang diangkat: urgensi pengambilan keputusan berbasis data untuk menyusun kebijakan penerimaan mahasiswa yang efektif, serta waktu eksekusi yang lama pada dataset besar seperti data SNBT 2023, yang mencakup ribuan program studi dengan variabel kompleks. Jumlah cluster ditentukan menggunakan metode elbow, membagi data ke dalam tiga kategori: low, medium, dan high. Evaluasi clustering dilakukan dengan metrik silhouette score, yang menunjukkan bahwa Cluster 0 (low) memiliki kualitas terbaik dengan nilai silhouette score tertinggi. Untuk mempercepat proses analisis, diterapkan parallel computing menggunakan library joblib, scikit learn,* dan *multiprocessing, yang terbukti mengurangi waktu eksekusi secara signifikan dibandingkan metode konvensional. Dengan rata-rata nilai silhouette score sebesar 0,684816, hasil penelitian ini menunjukkan kualitas pengelompokan yang baik. Temuan ini memberikan wawasan penting bagi perguruan tinggi dalam memahami pola keketatan program studi dan mendukung perancangan kebijakan penerimaan mahasiswa berbasis data yang lebih efektif dan efisien.*

Kata Kunci *– Keketatan Program Studi; K-Means Clustering; SNBT 2023; Parallel Computing; Silhouette Score.*

Parallel Computing in K-Means *Cluster*ing for the Analysis of Study Program Competitiveness in SNBT 2023

ABSTRACT *– This study aims to analyze the competitiveness of study programs in the 2023 SNBT dataset using the Knowledge Discovery in Databases (KDD) method and K-Means Clustering algorithm. The competitiveness of study programs is measured by the ratio between the number of applicants and available slots, reflecting the level of competition and popularity of the programs. Two main issues are addressed: the urgency of data-driven decision-making for formulating effective student admission policies and the lengthy execution time on large datasets such as the 2023 SNBT data, which includes thousands of study programs with complex variables. The number of clusters was determined using the elbow method, dividing data into three categories: low, medium, and high. Clustering evaluation was conducted using the silhouette score metric, revealing that Cluster 0 (low) demonstrated the best quality with the highest silhouette score. To accelerate the analysis process, parallel computing was implemented using joblib, scikit-learn, and multiprocessing library, significantly reducing execution time compared to conventional methods. With an average silhouette score of 0.684816, the results indicate good clustering quality. These findings provide valuable insights for universities in understanding the competitiveness patterns of study programs and support the development of more effective and efficient data-driven student admission policies.*

Keywords *– Program Study Competitiveness; K-Means Clustering; SNBT 2023; Parallel Computing; Silhouette Score.*

1. PENDAHULUAN

Keketatan program studi di perguruan tinggi merupakan indikator penting dalam mengevaluasi tingkat persaingan dan popularitas suatu program studi di Indonesia. Dalam konteks Seleksi Nasional Berdasarkan Tes (SNBT), keketatan ini mencerminkan rasio antara jumlah pendaftar dan daya tampung program studi. Program studi dengan tingkat keketatan tinggi menunjukkan tingginya minat pendaftar sekaligus persaingan ketat untuk diterima. Fenomena ini memberikan wawasan berharga bagi perguruan tinggi dalam merancang strategi penerimaan mahasiswa dan pengelolaan kapasitas program studi. Selain itu, meningkatnya jumlah pendaftar setiap tahunnya menambah kompleksitas analisis keketatan program studi, yang mencakup data seperti daya tampung, peminat, program studi, dan keketatan. Oleh karena itu, diperlukan pengolahan data yang efisien dan akurat untuk mendukung pengambilan keputusan berbasis data.

Pola minat pendaftar dan distribusi daya tampung sangat penting bagi institusi pendidikan dan pemerintah untuk menyusun kebijakan penerimaan mahasiswa yang lebih efektif. Tanpa analisis yang mendalam, sulit untuk mengoptimalkan pengelolaan program studi sesuai dengan kebutuhan dan minat pasar pendidikan. Data SNBT 2023 mencakup ribuan program studi dengan variabel seperti daya tampung, jumlah peminat, dan tingkat keketatan.

Proses *clustering* menggunakan *K-Means* pada *dataset* besar dapat memakan waktu lama jika dilakukan secara sekuensial. Implementasi *parallel computing* relevan untuk mempercepat analisis, terutama ketika jumlah iterasi *clustering* dan ukuran *dataset* sangat besar. Metode *clustering* merupakan pendekatan yang sering digunakan untuk menganalisis dan mengelompokkan data dengan karakteristik serupa [1]. Salah satu metode yang populer adalah *K-Means clustering*, yang mempartisi data ke dalam beberapa cluster berdasarkan kesamaan karakteristik. Dalam konteks analisis keketatan program studi, metode ini memungkinkan pengelompokan program studi berdasarkan daya tampung, jumlah peminat, dan tingkat keketatan untuk mengidentifikasi pola distribusi minat dan persaingan [2].

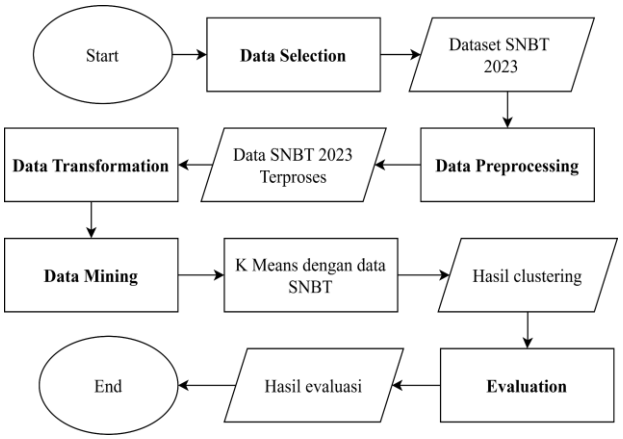
Penelitian oleh Quraata A'yuni et al. mengevaluasi pola penerimaan mahasiswa penerima beasiswa Bank Indonesia menggunakan analisis *clustering* dengan RapidMiner. Hasil penelitian menunjukkan lima *cluster* dengan karakteristik penerima beasiswa yang beragam berdasarkan parameter seperti program studi, IPK, dan jenjang pendidikan[3]. Penelitian lain oleh Muhammad Azzam Al Fauzie et al. menggunakan metode *K-Means* untuk menganalisis siswa aktif ekstrakurikuler. Hasilnya menunjukkan pola yang dapat digunakan untuk merekomendasikan pembelajaran akademik [4]. Sementara itu, Haris et al. menggunakan *K-Means* untuk menentukan besaran Uang Kuliah Tunggal (UKT) berdasarkan kondisi sosial ekonomi calon mahasiswa [5]. Peneliti lain yaitu Dwi Fitri dan Wawan Joko menggunakan *K-Means* dalam memilih strategi promosi penerimaan mahasiswa baru [6]. Keempat penelitian ini menunjukkan potensi metode *clustering* dalam menganalisis data pendidikan

Analisis keketatan program studi pada skala nasional sangat penting untuk mendukung perumusan kebijakan penerimaan mahasiswa yang lebih efektif. Dengan memahami pola minat pendaftar dan distribusi daya tampung, institusi pendidikan dan pemerintah dapat menyusun strategi yang lebih terarah dalam mengelola kapasitas program studi dan meningkatkan kualitas pendidikan tinggi di Indonesia.

Penelitian ini bertujuan untuk menganalisis keketatan program studi pada data SNBT tahun 2023 dengan menggunakan metode *Knowledge Discovery in Databases* (KDD) dan *K-Means clustering*. Jumlah *cluster* ditentukan menggunakan metode elbow, dengan pembagian menjadi tiga kategori utama: *low*, *medium*, dan *high*. Evaluasi *clustering* dilakukan menggunakan metrik *silhouette score* untuk memastikan validitas dan kualitas hasil pengelompokan. Implementasi *parallel computing* menggunakan *library joblib, scikit-learn,* dan *multiprocessing* diterapkan untuk mempercepat waktu eksekusi analisis. Hasil dari penelitian ini diharapkan dapat memberikan wawasan yang mendalam bagi perguruan tinggi dan pemangku kepentingan dalam pengelolaan program studi berbasis data.

2. METODE DAN BAHAN

Penelitian ini menerapkan proses *clustering* untuk mengelompokkan tingkat keketatan program studi berdasarkan data SNBT tahun 2023 menggunakan metode *K-Means*. *Dataset* yang digunakan mencakup program studi dari berbagai universitas di Indonesia, dengan atribut seperti daya tampung, peminat, dan keketatan program studi. Penelitian ini dilakukan menggunakan bahasa pemrograman *Python*, dengan *library joblib, sklearn,* dan *multiprocessing* untuk mempercepat proses *parallel computing*. Pendekatan utama yang digunakan adalah metodologi *Knowledge Discovery in Databases* (KDD), Data yang dianalisis dalam penelitian ini merupakan data sekunder yang diambil dari platform Kaggle. Gambar 1 merupakan diagram alir dari alur penelitian ini :



Gambar 1. Diagram alir penelitian.

Knowledge Discovery in Databases (KDD) adalah proses menemukan informasi baru yang berguna dari data dalam basis data melalui pengolahan data yang terstruktur [7]. Tujuannya adalah mengubah data mentah menjadi pengetahuan yang bermanfaat untuk pengambilan keputusan [8]. KDD sendiri terdiri dari beberapa tahap [9], yaitu:

1. *Data Selection*

Tahap ini bertujuan untuk memilih data yang relevan yang mendukung analisis [10]. Penulis memilih data sekunder SNBT 2023 dari sumber *Kaggle*, yang mencakup atribut universitas, nama program studi, daya tampung, dan jumlah peminat. Pemilihan data ini dilakukan untuk memastikan relevansi dengan tujuan analisis mengenai keketatan program studi penjelasan ini ditampilkan lebih jelas pada Tabel 1.

2. *Data Preprocessing*

Tahap ini bertujuan untuk mempersiapkan data agar siap digunakan dalam proses *cluster*ing [11]. Penulis menghapus beberapa data yang tidak relevan serta menangani nilai yang tidak ada.

3. *Data Transformation*

Tahap ini bertujuan untuk mengubah data yang telah diproses agar lebih cocok atau lebih mudah digunakan [12]. Penulis melakukan agregasi pada kolom daya tampung dan peminat menjadi keketatan.

4. *Data Mining*

Pada tahap ini, dilakukan proses *clustering* menggunakan metode *K-Means*. Implementasi algoritma *K-Means* menggunakan *library joblib, sklearn,* dan *multiprocessing*. Berikut pseudocode untuk algoritma *K-Means* pada penelitian ini.

```
Fungsi SekuentialClustering(input_data, n_cluster)
{
  Input:
    Input_data = Dataset dengan kolom 'keketatan'
    n_cluster = Jumlah cluster

  Output:
    waktu_eksekusi = Waktu eksekusi clustering

  Proses:
    - Menonaktifkan paralelisasi OpenMP
    - Start timer
    - Ambil kolom 'keketatan' dari Input_data
    - Terapkan K-Means clustering dengan jumlah n_cluster = 3
    - Simpan hasil clustering ke dalam kolom 'cluster' pada Input_data
    - Membuat mapping cluster ke kategori (low, medium, high)
    - Kelompokkan berdasarkan kategori keketatan
    - Sorting hasil berdasarkan keketatan tertinggi
    - Stop timer
    - Tampilkan waktu_eksekusi
}

Fungsi ParallelClustering(input_data, n_cluster, metode)
{
  Input:
    input_data = Dataset dengan kolom 'keketatan'
    n_cluster = Jumlah cluster
    metode = Multiprocessing, Joblib, atau Scikit-learn

  Output:
    waktu_eksekusi = Waktu eksekusi pengolahan data

  Proses:
  1. Clustering dengan K-Means (pada semua metode)
    - Mulai timer
    - Inisialisasi model KMeans dengan n_cluster
    - Fit dataset menggunakan 'KMeans.fit()' dan dapatkan label cluster untuk setiap data
    - Tentukan kategori cluster ('low', 'medium', 'high') berdasarkan centroid
    - Simpan kategori hasil clustering dalam dataset

  2. Pengolahan kategori dengan metode parallel
  metode = Multiprocessing:
    - Mulai timer
    - Gunakan 'multiprocessing.Pool()' untuk menjalankan proses secara parallel
    - Gunakan 'pool.map()' untuk membagi pekerjaan ke beberapa proses
    - Gabungkan hasil dari semua proses
    - Hentikan timer dan tampilkan waktu_eksekusi
  metode = Joblib:
    - Mulai timer
    - Gunakan 'Parallel(n_jobs=-1)' untuk menggunakan semua core CPU yang tersedia
    - Gunakan 'delayed()' untuk mendistribusikan tugas ke beberapa core CPU
    - Gabungkan hasil dari semua core
    - Hentikan timer dan tampilkan waktu_eksekusi
  metode = Scikit-learn:
    - Mulai timer
    - Gunakan 'KMeans.fit()' (multi-threading otomatis)
    - Mapping cluster ke kategori ('low', 'medium', 'high')
    - Hentikan timer dan tampilkan waktu_eksekusi
```

```
}
```

5. Evaluation

Tahap ini melakukan evaluasi hasil *clustering* menggunakan metode *silhouette score*. Metode ini bertujuan untuk mengukur seberapa baik data dalam setiap *cluster* terkelompokkan dan sejauh mana masing-masing data cocok dengan *cluster* yang dimilikinya dibandingkan dengan *cluster* lainnya [14].

3. HASIL DAN PEMBAHASAN

Data Selection

Data yang digunakan dalam penelitian ini mencakup 126 universitas di Indonesia dengan total 1.225 program studi yang berbeda, menghasilkan 4.784 baris data. Atribut yang digunakan untuk proses *cluster*ing meliputi universitas, program studi, daya tampung, dan jumlah peminat seperti yang terlihat pada Tabel 1.

Tabel 1. *Data selection*

No	Universitas	Nama Prodi	Daya Tampung	Peminat
1	UNIVERSITAS SYIAH KUALA	PENDIDIKAN DOKTER HEWAN	84	548
2	UNIVERSITAS SYIAH KUALA	TEKNIK SIPIL	98	587
3	UNIVERSITAS SYIAH KUALA	TEKNIK MESIN	42	231
…	….	….	….	….
2997	UNIVERSITAS HASANUDDIN	TEKNIK INFORMATIKA	60	2019
2998	UNIVERSITAS HASANUDDIN	FISIOTERAPI	40	400
2999	UNIVERSITAS HASANUDDIN	TEKNIK LINGKUNGAN	60	401
…	….	….	….	….
4760	UNIVERSITAS ISLAM NEGERI DATOKARAMA PALU	INFORMATIKA	45	31
4760	UNIVERSITAS ISLAM NEGERI DATOKARAMA PALU	ARSITEKTUR	18	5
4760	UNIVERSITAS ISLAM NEGERI DATOKARAMA PALU	SISTEM INFORMASI	36	24

Data Preprocessing

Tahap ini dilakukan melalui beberapa langkah untuk mempersiapkan data agar siap digunakan dalam proses analisis. Langkah-langkah tersebut meliputi mengisi nilai yang hilang (*missing values*) dengan nilai rata-rata (*mean*) dari kelompok data terkait, menghapus data yang tidak relevan atau tidak diperlukan untuk analisis. Data yang telah diproses berjumlah 4.760 baris.

Data Transformation

Tahap *data transformation* ini dilakukan proses agregasi untuk menghitung keketatan setiap program studi. Agregasi ini bertujuan untuk menggabungkan informasi terkait jumlah peminat dan daya tampung program studi, yang kemudian dihitung menggunakan rumus keketatan, yaitu pada rumus (2) di bawah. Tabel 2 menunjukan contoh dari kolom keketatan yang sudah dihitung.

$$Keketatan = \frac{Jumlah\ peminat}{Daya\ Tampung} \quad (2)$$

Tabel 2. *Data Transformation*

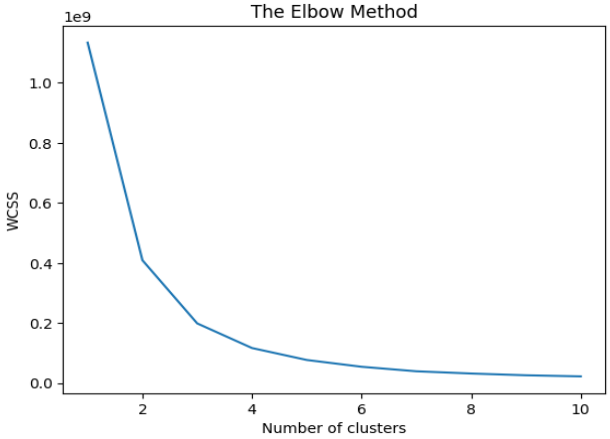
Universitas	Program Studi	Daya Tampung	Peminat	Keketatan
UNIVERSITAS INDONESIA	KRIMINOLOGI	18	1597	88.72222222
UNIVERSITAS INDONESIA	SOSIOLOGI	24	379	15.79166667
UNIVERSITAS INDONESIA	ILMU KESEJAHTERAAN SOSIAL	24	558	23.25
UNIVERSITAS INDONESIA	ANTROPOLOGI SOSIAL	21	490	23.33333333
UNIVERSITAS INDONESIA	ILMU EKONOMI	59	878	14.88135593

Data Mining

Berikut adalah langkah-langkah analisis program studi menggunakan algoritma *K-Means Clustering*:

a. Menentukan jumlah *cluster*

Pada penelitian ini, untuk menentukan jumlah *cluster* yang optimal, digunakan metode *Elbow*. Metode ini membantu dalam memilih jumlah cluster yang paling tepat berdasarkan grafik yang menunjukkan perubahan dalam total *within-cluster sum of squares* (WCSS). WCSS mengukur seberapa dekat titik-titik data di dalam satu *cluster* dengan *centroid*-nya, dan grafik *Elbow* menggambarkan penurunan nilai WCSS seiring bertambahnya jumlah *cluster* [15]. Penentuan jumlah *cluster* dilakukan menggunakan bahasa pemrograman *Python* pada *Google Colab*, dengan parameter *random_state* diatur ke 42 untuk memastikan hasil yang konsisten. Gambar 2 berikut menunjukkan hasil penentuan jumlah cluster yang digunakan dalam penelitian ini.



Gambar 2. Hasil penentuan jumlah *cluster* menggunakan metode *Elbow*

Grafik percobaan dengan metode *Elbow* di atas menunjukkan adanya penyusutan dan lengkungan yang signifikan. Oleh karena itu, nilai K maksimum (jumlah *cluster* yang optimal) ditentukan pada titik lokasi yang ideal, yaitu tiga *cluster*. Setelah memeriksa hasil diagram dari metode *Elbow*, pengelompokan menggunakan *K-Means* dilakukan dengan membagi data menjadi tiga *cluster* yang berbeda.

b. Menentukan pusat *cluster*

Setelah jumlah *cluster* ditentukan menggunakan metode *Elbow*, langkah berikutnya adalah menentukan pusat *cluster* atau *centroid* awal. Pada penelitian ini, digunakan *library scikit-learn* untuk mengimplementasikan algoritma *K-Means Cluster*ing. Penentuan *centroid* awal dilakukan secara random (default) oleh *library*, yang berarti titik-titik awal *centroid* dipilih secara acak dari kumpulan data yang digunakan.

c. Jarak objek ke *centroid* menggunakan persamaan *Euclidean*

Pada tahap berikutnya, dilakukan penghitungan jarak antara setiap titik data dan ketiga *centroid* yang telah ditentukan. Penghitungan jarak ini dilakukan menggunakan metode *Euclidean distance*, yang mengukur seberapa jauh setiap titik data dari posisi *centroid* pada ruang fitur. Proses ini memungkinkan data dikelompokkan berdasarkan kedekatannya dengan *centroid* yang ada, sehingga setiap titik data akan dikelompokkan ke dalam *cluster* yang memiliki jarak *Euclidean* terkecil dari *centroid*-nya. Hal ini dilakukan secara iteratif sampai posisi *centroid* stabil dan tidak berubah lagi. Tabel 3 menampilkan jarak setiap objek ke setiap *cluster*.

Tabel 3. Jarak data dengan *cluster*

No	Universitas	Nama Prodi	Jarak *Cluster* 0	Jarak *Cluster* 1	Jarak *Cluster* 2	Jarak Terdekat	*Cluster*
1	UNIVERSITAS SYIAH KUALA	PENDIDIKAN DOKTER HEWAN	3.5476108	33.152013	8.32750852	3.54761085	0
2	UNIVERSITAS SYIAH KUALA	TEKNIK SIPIL	3.01359724	33.686027	8.86152213	3.01359724	0
3	UNIVERSITAS SYIAH KUALA	TEKNIK MESIN	2.52380133	34.175822	9.35131804	2.52380133	0

No	Universitas	Nama Prodi	Jarak *Cluster* 0	Jarak *Cluster* 1	Jarak *Cluster* 2	Jarak Terdekat	*Cluster*
…	….	….	….	….	….	….	….
2380	UNIVERSITAS HASANUDDIN	TEKNIK INFORMATIKA	30.6738013	6.0258229	18.7986819	6.0258229	1
2381	UNIVERSITAS HASANUDDIN	FISIOTERAPI	7.02380133	29.675822	4.85131804	4.85131804	2
2382	UNIVERSITAS HASANUDDIN	TEKNIK LINGKUNGAN	3.70713466	32.992489	8.16798471	3.70713466	0
…	….	….	….	….	….	….	….
4758	UNIVERSITAS ISLAM NEGERI DATOKARAMA PALU	INFORMATIKA	2.28730978	38.986934	14.1624291	2.28730978	0
4759	UNIVERSITAS ISLAM NEGERI DATOKARAMA PALU	ARSITEKTUR	2.69842089	39.398045	14.5735402	2.69842089	0
4760	UNIVERSITAS ISLAM NEGERI DATOKARAMA PALU	SISTEM INFORMASI	2.30953200	39.009156	14.1846513	2.30953200	0

d. Clustering

Tahap ini dilakukan dengan menggunakan 2 jenis *clustering*.

1. Sequential

Jenis pertama, proses *cluster*ing dilakukan secara sekuensial menggunakan algoritma *K-Means* berdasarkan keketatan yang menggambarkan tingkat kepadatan atau tingkat keketatan suatu program studi menggunakan *library sklearn*. Untuk eksekusi secara sekuensial, variabel lingkungan *OMP_NUM_THREADS* di set menjadi "1", yang membatasi algoritma *K-Means* untuk hanya menggunakan satu inti CPU, tanpa melakukan paralelisasi menggunakan *OpenMP*. Sehingga, seluruh proses ini dilakukan satu per satu tanpa pemrosesan *parallel*.

2. Parallel

Pemrograman *parallel* merupakan teknik pemrograman yang dapat mengeksekusi beberapa tugas secara bersamaan dengan memanfaatkan beberapa inti (*core*) pada *processor*. Teknik ini digunakan untuk meningkatkan efisiensi dan performa program, terutama untuk komputasi yang memerlukan banyak sumber daya. Adapun *library* yang digunakan yaitu *joblib, scikit-learn* dan *multiprocessing* untuk melihat perbandingan waktu eksekusinya.

a) ***Joblib***

Pada *library joblib*, *clustering* dijalankan secara *parallel* yang membuat pemrosesan data menjadi lebih cepat dibandingkan. *Joblib* digunakan untuk memproses kategori secara *parallel*. Fungsi *parallel* digunakan melalui parameter *n_jobs=-1* yang berfungsi untuk menjalankan pemrosesan pada semua *core CPU* yang tersedia dengan menggunakan fungsi *parallel* dan *delayed*, dimana kode menjalankan fungsi *process_category* untuk masing-masing kategori secara bersamaan.

b) Scikit-learn

Selanjutnya yaitu *library scikit-learn* atau *sklearn*. Algoritma *K-Means* di *scikit-learn* secara internal menggunakan *multithreading* untuk beberapa operasi dalam proses pelatihan model, terutama dalam menghitung jarak data dan memperbarui posisi *centroid*. Ketika *K-Means* mencoba untuk menemukan *cluster* yang optimal, ia harus menghitung jarak dari setiap titik data ke masing-masing *centroid*. Proses ini diparalelkan karena setiap perhitungan jarak dapat dilakukan secara independen untuk setiap titik data. Pada *library scikit-learn*, *multi-threading* terjadi secara otomatis tanpa harus mengkonfigurasi parameter *n_jobs* (karena *n_jobs* pada *scikit-learn* versi 0.23 pada algoritma *cluster.K-Means* sudah didepresiasi). Dengan ini, *scikit-learn* dapat menggunakan banyak inti CPU (CPU *cores*) yang tersedia untuk mempercepat komputasi.

c) Multiprocessing

Terakhir menggunakan *library multiprocessing*. *Library* ini menggunakan *multiprocessing.pool* yang akan membuat *worker processes*, tujuannya untuk menjalankan beberapa proses secara *parallel*. *Pool* ini membagi tugas pemrosesan ke beberapa proses terpisah, yang mengurangi waktu yang dibutuhkan untuk memproses data yang sangat banyak.

Hasil *Clustering K-Means*

Tabel 4. Label *cluster*

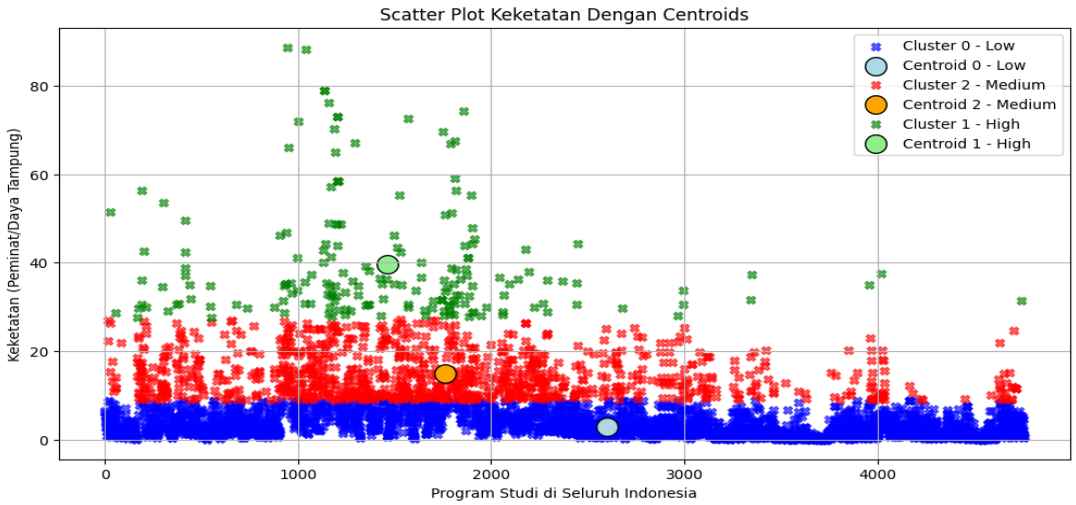
CLUSTER	LABEL
Cluster 0	*Low*
Cluster 1	*High*
Cluster 2	*Medium*

Tabel 4 menjelaskan bahwa *cluster* 0 diberi label *Low*, *Cluster* 1 diberi label *High*, dan *Cluster* 2 diberi label *Medium*. Dengan demikian, pengelompokan ini memberikan gambaran yang jelas mengenai tingkat persaingan di setiap program studi, yang dapat digunakan untuk analisis lebih lanjut terkait keketatan penerimaan mahasiswa baru di setiap universitas.

Tabel 5. Jumlah data pada setiap *cluster*

CLUSTER	JUMLAH
Cluster 0	3575
Cluster 1	183
Cluster 2	1002
Total *Cluster*	4760

Tabel 5 menjelaskan jika dilakukan pengelompokan data SNBT berdasarkan keketatan program studi di setiap universitas menggunakan metode *K-Means*. Hasil dari pengelompokan ini menghasilkan tiga *cluster* yang menggambarkan tingkat keketatan yang berbeda antara program studi.



Gambar 3. Hasil *clustering* menggunakan *K-Means*

Gambar 3 menunjukan adanya sumbu X *(Horizontal)* yang merujuk pada daftar program studi yang ada didalam *dataset*, kemudian sumbu Y *(Vertikal)* yang menyatakan keketatan berdasarkan peminat dan daya tampung dari suatu program studi. Terdapat tiga lingkaran menunjukan *centroid* atau titik pusat yang menyatakan rata-rata dari data yang ada dalam *cluster* tersebut. Berikut adalah penjelasan mengenai analisis *cluster* yang tercantum pada Gambar 3:

1. *Cluster* 0 (*Low*)

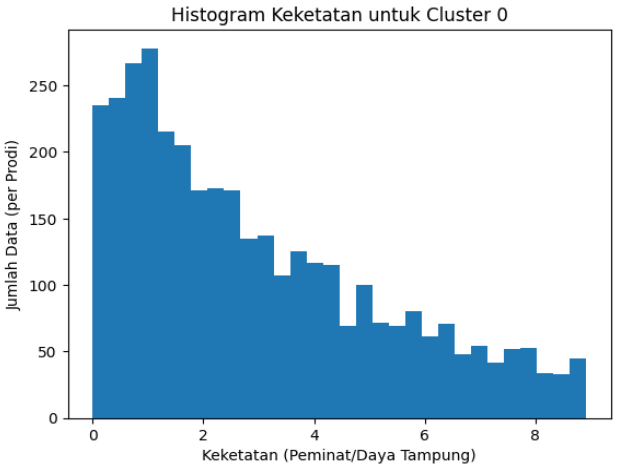
Cluster dengan warna biru ini menunjukan program studi dengan keketatan rendah yang memiliki daya tampung lebih tinggi atau peminat yang lebih sedikit (gap antara jumlah peminat dan daya tampung dekat). Titik pusat ditandai dengan *centroid* 0 berwarna *light blue.*

2. *Cluster* 1 (*High*)

Cluster berwarna hijau menunjukan program studi dengan keketatan tinggi cenderung memiliki daya tampung yang lebih terbatas atau peminat yang lebih banyak (gap antara jumlah peminat dan daya tampung jauh). Titik pusat ditandai dengan *centroid* 1 berwarna *light green.*

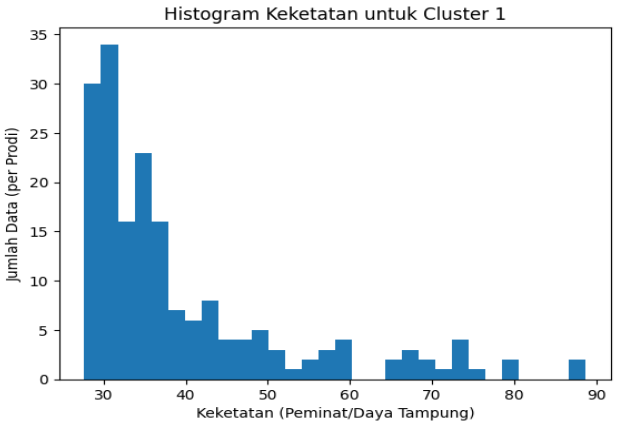
3. *Cluster* 2 (*Medium*)

Cluster berwarna merah menunjukan bahwa program studi dengan keketatan sedang terletak di antara kedua *cluster* lainnya, dengan daya tampung dan peminat yang lebih seimbang. Titik pusat ditandai dengan *centroid* 2 berwarna *orange.*



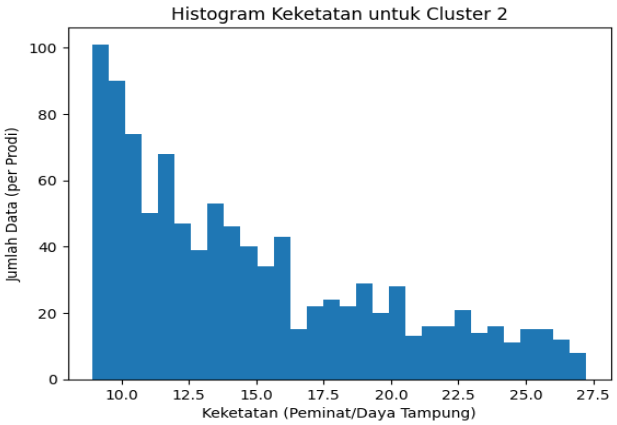
Gambar 4. Histogram keketatan pada *cluster* 0.

Gambar 4 menunjukkan sebaran data pada *cluster* 0 yang memiliki tingkat keketatan terendah. Rentang keketatan dalam *cluster* ini berkisar antara 0 hingga 8.913043, menandakan variabilitas yang relatif rendah pada data tersebut.



Gambar 5. Histogram keketatan pada *cluster* 1.

Gambar 5 menunjukkan sebaran data pada *cluster* dengan rentang keketatan antara 27.666667 hingga 88.722222. *Cluster* ini mencerminkan tingkat keketatan yang lebih tinggi dibandingkan *cluster* sebelumnya, dengan data yang memiliki variasi lebih besar dalam nilai keketatan.



Gambar 6. Histogram keketatan pada *cluster* 2.

Gambar 6 menunjukkan sebaran data pada *cluster* dengan rentang keketatan antara 8.927273 hingga 27.238095. *Cluster* ini menunjukkan tingkat keketatan yang sedang, dengan variasi nilai keketatan yang lebih seimbang jika dibandingkan dengan *cluster* dengan rentang lebih tinggi atau lebih rendah.

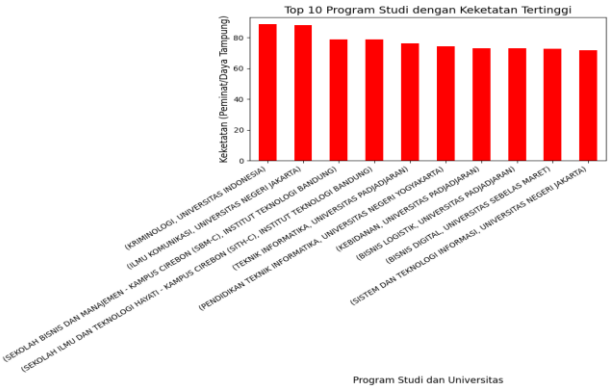
Tabel 6. Hasil cluster menggunakan *parallel computing*

	LIBRARY	**WAKTU EKSEKUSI /DETIK**
SEQUENTIAL	-	1.3876
PARALEL	*JOBLIB*	0.0279
	SCIKIT-LEARN	0.0787
	MULTIPROCESSING	0.1203

Dari Tabel 6 dapat disimpulkan bahwa *library joblib* menghasilkan waktu eksekusi yang paling cepat dibandingkan dengan *sequential* dan *library parallel* lainnya. Hasil waktu eksekusinya yang cepat mengindikasikan bahwa *joblib* sangat optimal dalam memanfaatkan berbagai *core processor*. *Joblib* menggunakan teknik serialisasi yang lebih cepat dan efisien dibandingkan dengan *library* lainnya. Hal ini membuat data dapat diproses lebih cepat, karena proses untuk memindahkan data antar *thread* atau *processor* lebih ringan. *Joblib* juga lebih efisien dalam hal pengelolaan sumber daya, *library* ini mengelola jumlah *thread* dan *processor* yang digunakan secara otomatis, mengurangi *overhead* yang terkait dengan pembagian tugas di banyak *processor*.

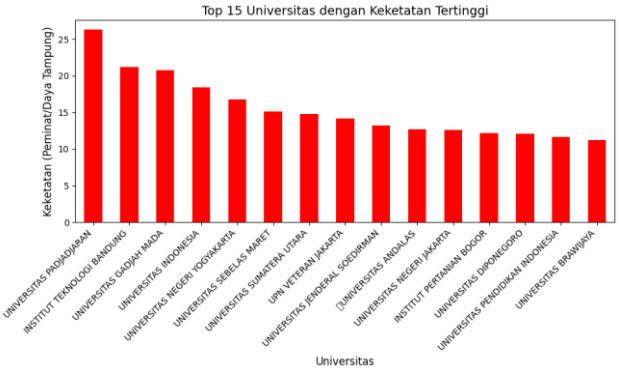
Library Multiprocessing memiliki waktu eksekusi lebih lama dibandingkan dengan *library parallel* lain karena adanya *overhead* yang terjadi dalam proses manajemen multi proses. *Multiprocessing* bekerja dengan membagi tugas ke beberapa proses terpisah.

Setiap proses berjalan secara independen, dan untuk itu data perlu dibagi dan disalin ke setiap proses secara terpisah. Jika tugas yang diberikan relatif kecil atau tidak cukup kompleks untuk memanfaatkan beberapa proses, maka waktu yang dibutuhkan untuk menyiapkan dan manajemen proses (*overhead*) ini akan lebih besar daripada keuntungan yang diperoleh dari paralelisasi. Jadi, *library multiprocessing* ini hanya cocok digunakan untuk *dataset* yang sangat besar.



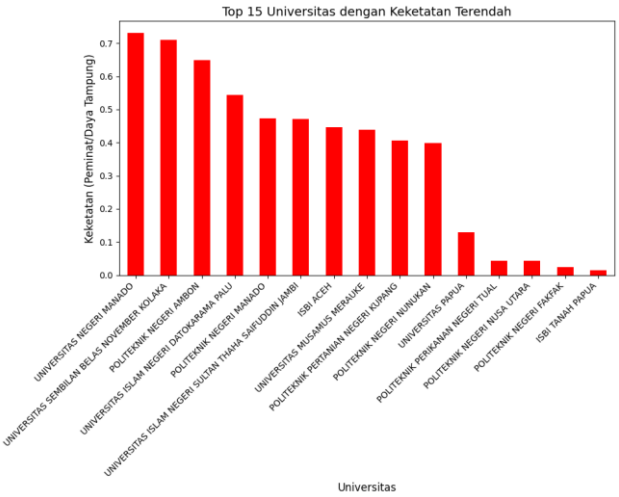
Gambar 7. Program studi dengan keketatan paling tinggi.

Grafik pada gambar 7 ini menampilkan 10 program studi dengan tingkat keketatan tertinggi di Indonesia. Program studi ini memiliki rasio yang mencerminkan tingginya persaingan masuk akibat tingginya minat calon mahasiswa terhadap daya tampung yang terbatas.



Gambar 8. Universitas dengan keketatan tertinggi.

Grafik pada gambar 8 menunjukkan rasio rata-rata keketatan dari 15 universitas dengan tingkat persaingan tertinggi di Indonesia. Rasio ini dihitung berdasarkan perbandingan antara jumlah peminat terhadap daya tampung yang tersedia untuk setiap program studi di universitas tersebut. Universitas ini memiliki rasio keketatan yang tinggi dan menunjukkan persaingan yang sangat ketat untuk mendapatkan kursi di program-program studi yang ditawarkan.



Gambar 9. Universitas dengan keketatan terendah.

Grafik pada gambar 9 menunjukkan rasio rata-rata keketatan dari 15 universitas dengan tingkat persaingan terendah di Indonesia. Rasio ini merepresentasikan jumlah peminat yang rendah dibandingkan daya tampung yang tersedia untuk setiap program studi di universitas tersebut.

Evaluation

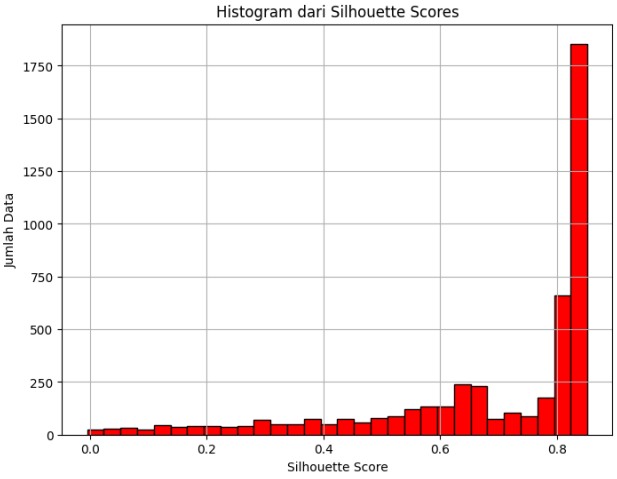
Tahap ini melakukan evaluasi hasil *clustering* menggunakan metode *silhouette score*. Metode ini bertujuan untuk mengukur seberapa baik data dalam setiap *cluster* terkelompokkan dan sejauh mana masing-masing data cocok dengan *cluster* yang dimilikinya dibandingkan dengan *cluster* lainnya. Nilai *silhouette score* berkisar antara -1 hingga 1, di mana nilai mendekati 1 menunjukkan pengelompokan yang baik, nilai mendekati 0 menunjukkan data berada di batas antara dua *cluster*, dan nilai negatif menunjukkan data kemungkinan salah dikelompokkan. Evaluasi ini membantu menentukan kualitas dan validitas hasil *clustering*. Gambar 10 memperlihatkan hasil silhouette s*core* pada setiap *cluster*.

```
   cluster  silhouette_score
0        0          0.756205
1        1          0.450042
2        2          0.472987

Rata-rata Silhouette Score: 0.6848159647238742
```

Gambar 10. Hasil uji menggunakan metode *silhouette* s*core*

Berdasarkan hasil pengujian pada *dataset* SNBT tahun 2023 yang terdiri dari 4760 data uji, kualitas *cluster* diuji pada *Cluster* 0 hingga *Cluster* 2. Dari hasil evaluasi, *Cluster* 0 memiliki kualitas pengelompokan terbaik dengan nilai *Silhouette Score* sebesar 0,756205. Rata-rata nilai *Silhouette Score* untuk seluruh *cluster* dari *Cluster* 0 hingga *Cluster* 2 adalah 0,6848156, yang menunjukkan bahwa pengelompokan data secara keseluruhan memiliki kualitas yang baik dan dapat diandalkan untuk analisis lebih lanjut. Gambar 11 di bawah memperlihatkan bahwa ada lebih dari 1750 data yang *silhouette score* nya lebih dari 0,8.



Gambar 11. Histogram *silhouette score*

4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang dilakukan, dapat disimpulkan bahwa penelitian ini berhasil membentuk tiga *cluster* menggunakan metode *Elbow* dalam algoritma *K-Means Clustering*. Hasil *clustering* menunjukkan bahwa *Cluster* 0 dengan label *Low* memiliki 3575 data, *Cluster* 1 dengan label *High* memiliki 183 data, dan *Cluster* 2 dengan label *Medium* memiliki 1002 data.

Perbandingan dilakukan antara berbagai *library* dan waktu eksekusi secara sequential dan secara *parallel computing*. Proses *cluster*ing dilakukan menggunakan bahasa pemrograman *Python*, dengan memanfaatkan metode *parallel computing* menggunakan *library* seperti *joblib*, *scikit-learn* (*sklearn*), dan *multiprocessing* untuk meningkatkan efisiensi komputasi. Waktu eksekusi yang dihasilkan adalah 0,0279 detik menggunakan *joblib*, 0,0787 detik menggunakan sklearn, dan 0,1203 detik dengan *multiprocessing*, sementara eksekusi tanpa *parallel computing* membutuhkan waktu 1,3876 detik. Hasil ini menunjukkan bahwa penggunaan parallel *computing* dapat mempercepat waktu eksekusi, dimana *joblib* menunjukkan performa terbaik dalam penelitian ini.

Evaluasi hasil *cluster*ing dilakukan menggunakan metode Silhouette S*core*, yang menunjukkan bahwa *Cluster* 0 memiliki nilai *Silhouette* S*core* tertinggi dibandingkan *cluster* lainnya. Rata-rata nilai *Silhouette Score* untuk seluruh *cluster* adalah 0,684816, yang menunjukkan bahwa kualitas pengelompokan yang dilakukan berada pada kategori yang baik dan dapat digunakan untuk analisis lebih lanjut.

DAFTAR PUSTAKA

[1] Sekar Setyaningtyas, B. Indarmawan Nugroho, and Z. Arif, "Tinjauan Pustaka Sistematis: Penerapan Data Mining Teknik *Cluster*ing Algoritma *K-Means*," *J. Teknoif Tek. Inform. Inst. Teknol. Padang*, vol. 10, no. 2, pp. 52–61, 2022, doi: 10.21063/jtif.2022.v10.2.52-61.

[2] A. Septiarini, I. A. Thaher, and N. Puspitasari, "Pengelompokan Kualitas Kinerja Pegawai Menggunakan Metode K-Means Clustering," *Komputika J. Sist. Komput.*, vol. 11, no. 2, pp. 131–141, 2022.

[3] Q. A'yuni, A. Nazir, L. Handayani, and I. Afrianty, "Penerapan Algoritma *K-Means Cluster*ing untuk Mengetahui Pola Penerima Beasiswa Bank Indonesia (BI)," *J. Comput. Syst. Informatics*, vol. 4, no. 3, pp. 530–539, 2023.

[4] M. Azzam Al Fauzie and J. Akhir Putra, "*Cluster*ing Data Menggunakan Metode *K-Means* untuk Rekomendasikan Pembelajaran Akademik bagi Siswa Aktif dalam Ekstrakurikuler," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 1, pp. 642–648, 2023, doi: 10.30865/klik.v4i1.1116.

[5] Haris Kurniawan, Sarjon Defit, and Sumijan, "Data Mining Menggunakan Metode *K-Means Cluster*ing Untuk Menentukan Besaran Uang Kuliah Tunggal," *J. Appl. Comput. Sci. Technol.*, vol. 1, no. 2, pp. 80–89, 2020, doi: 10.52158/jacost.v1i2.102.

[6] Miliani, Dwi Fitri, and Wawan Joko Pranoto. "PENERAPAN K-MEANS CLUSTER DALAM MEMILIH STRATEGI PROMOSI PENERIMAAN MAHASISWA BARU." Jurnal Cahaya Mandalika ISSN 2721-4796 (online) (2024): 1665-1676.

[7] F. Nurdiyansyah and I. Akbar, "Implementasi Algoritma *K-Means* untuk Menentukan Persediaan Barang pada Poultry Shop," *J. Teknol. dan Manaj. Inform.*, vol. 7, no. 2, pp. 86–94, 2021.

[8] D. Damayanti, "Implementasi Algoritma C4. 5 Prediksi Produksi Komoditas Tanaman Perkebunan Berdasarkan Luas Lahan," *Tin Terap. Inform. Nusant.*, vol. 2, no. 10, pp. 571–579, 2022.

[9] A. Asmana, Y. A. Wijaya, and M. Martanto, "*Cluster*ing data calon siswa baru menggunakan metode *K-Means* di sekolah menengah kejuruan wahidin kota cirebon," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 552–559, 2022.

[10] H. A. Yanti, "Pengolahan data sederhana menggunakan R STUDIO," *Sienna*, vol. 2, no. 1, pp. 1–9, 2021.

[11] F. Alghifari and D. Juardi, "Penerapan Data Mining Pada Penjualan Makanan dan Minuman Menggunakan Metode Algoritma Na{\"\i}ve Bayes: Studi Kasus: Makan Barbeque Sepuasnya," *J. Ilm. Inform.*, vol. 9,

no. 02, pp. 75–81, 2021.

[12] S. Pujiono, R. Astuti, and F. M. Basysyar, "Implementasi Data Mining Untuk Menentukan Pola Penjualan Produk Menggunakan Algoritma *K-Means Cluster*ing," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 615–620, 2024.

[13] M. R. Alhapizi, M. Nasir, and I. Effendy, "Penerapan Data Mining Menggunakan Algoritma *K-Means Cluster*ing Untuk Menentukan Strategi Promosi Mahasiswa Baru Universitas Bina Darma Palembang," *J. Softw. Eng. Ampera*, vol. 1, no. 1, pp. 1–14, 2020.

[14] A. Fira, C. Rozikin, G. Garno, and others, "Komparasi Algoritma *K-Means* dan K-Medoids Untuk Pengelompokkan Penyebaran Covid-19 di Indonesia," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 133–138, 2021.

[15] S. Hendrawan, F. K. Sari Dewi, and Pranowo, "*Cluster*ing Evaluasi Dosen Universitas Atma Jaya Yogyakarta Menggunakan Metode *K-Means*," *J. Inform. Atma Jogja*, vol. 4, no. 1, pp. 1–8, 2023, doi: 10.24002/jiaj.v4i1.7436.