JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 10, No. 2, MEI, 2025, Pp. 145 – 157
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Analisis Cluster untuk Pengelompokan Kemampuan Penguasaan ICT Menggunakan K-Means dan *Autoencoder*

Daru Prasetyawan [(1)*], Rahmadhan Gatra [(2)]
Pusat Teknologi Informasi dan Pangkalan Data, UIN Sunan Kalijaga, Yogyakarta, Indonesia
e-mail : {daru.prasetyawan,rahmadhan.gatra}@uin-suka.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 16 Februari 2024, direvisi 14 Mei 2024, diterima 25 Juli 2024, dan dipublikasikan 31 Mei 2025.

Abstract

Information and Communication Technology (ICT) skills are essential in today's digital age. However, numerous new students possess varying levels of ICT proficiency and may lack the necessary skills expected by universities. ICT training is essential for enhancing students' ICT skills. Nevertheless, delivering the same training to all students proves to be less effective. Therefore, grouping students' ICT skills is crucial to ensure that the training provided aligns with the fundamental abilities of the students. Cluster analysis is a common method for grouping data. This study employs k-Means and autoencoder for cluster analysis, with autoencoder utilized to reduce data dimensions and k-Means to perform the clustering process. The Elbow method is utilized to identify the ideal number of clusters. The optimal number of clusters determined was three clusters. Model evaluation was conducted using the Silhouette coefficient and the Davies-Bouldin Index (DBI). The evaluation results revealed that the combination of k-Means and autoencoder yields superior performance compared to using k-Means alone, as evidenced by a higher Silhouette value and a lower DBI value.

Keywords: Clustering, K-Means, Autoencoder, ICT, Silhouette, Davies-Bouldin Index

Abstrak

Kemampuan ICT menjadi hal yang penting yang harus dikuasi oleh mahasiswa di era digital saat ini. Akan tetapi banyak mahasiswa baru yang memiliki kemampuan penguasaan ICT yang berbeda-beda, bahkan belum memiliki kemampuan yang dipersyaratkan oleh perguruan tinggi. Pelatihan ICT diperlukan untuk meningkatkan kemampuan ICT bagi mahasiswa. Akan tetapi, dengan memberikan pelatihan yang sama kepada semua mahasiswa menjadi kurang efektif. Oleh karena itu, pengelompokan kemampuan penguasaan ICT ini menjadi sangat penting agar pelatihan yang diberikan kepada mahasiswa sesuai dengan kemampuan dasar yang dimiliki mahasiswa. Analisis *cluster* merupakan salah satu cara yang sering digunakan dalam pengelompokan data. Penelitian ini menggunakan k-Means dan *autoencoder* untuk analisis *cluster*. *Autoencoder* digunakan untuk mereduksi dimensi data. Selanjutnya k-Means melakukan proses *clustering* data tersebut. Metode *Elbow* digunakan untuk menentukan jumlah *cluster* yang optimal. Jumlah *cluster* optimal yang diperoleh sebanyak tiga *cluster*. Evaluasi model dilakukan menggunakan *Silhoutte Cofficient* dan *Davies-Bouldin (DBI)*. Dari hasil evaluasi, diketahui bahwa kombinasi k-Means dan *autoencoder* menghasilkan kinerja yang lebih baik dibandingkan hanya dengan menggunakan k-Means saja, yang ditunjukan dengan *Silhoutte score* yang lebih tinggi dan nilai DBI yang lebih rendah.

Kata Kunci: *Clustering*, K-Means, *Autoencoder*, ICT, Silhouette, Davies-Bouldin Index

1. PENDAHULUAN

Teknologi Informasi atau Information Technologi (IT) menjadi salah satu kebutuhan utama untuk mencapai efisiensi dan efektifitas di dalam berbagai bidang. Bagi mahasiswa, teknologi informasi sangat berperan untuk mendukung kegiatan perkuliahan seperti dalam pengerjaan tugas, praktikum, dan proses belajar mengajar itu sendiri. Mahasiswa harus mampu belajar secara mandiri untuk meningkatkan pengetahuan dan keahlian tanpa bantuan dari orang lain, atau yang sering dikenal sebagai *self-direct learning*. Oleh karena itu, kemampuan penguasan teknologi informasi menjadi hal yang penting bagi mahasiswa, terlebih di era revolusi industri 4.0 yang

menuntut kreatifitas dan fleksibilitas kognitif. Mahasiswa dituntut untuk dapat berpikir dengan sudut pandang yang berbeda, serta mampu mempelajari hal-hal baru yang dapat mengasah kreatifitas.

Salah satu upaya untuk meningkatkan kemampuan penguasaan teknologi informasi adalah dengan pelatihan. Dengan pelatihan diharapkan dapat meningkatkan kemampuan penguasaan teknologi informasi bagi mahasiswa. Akan tetapi, dengan memberikan pelatihan yang sama kepada semua mahasiswa menjadi kurang efektif. Hal ini disebabkan oleh kemampuan dasar tentang teknologi informasi yang dimiliki mahasiswa tidak semua sama. Oleh sebab itu, perlakuan berbeda juga diperlukan untuk memberikan pelatihan terhadap mahasiswa tersebut berdasarkan kemampuan dasar yang sudah dimilikinya sebelumnya. Pengelompokan kemampuan penguasaan ICT ini menjadi tantangan sendiri untuk memberikan pelatihan secara tepat sesuai dengan kemampuan dasar yang dimiliki mahasiswa. Untuk menjawab tantangan tersebut. peneliti mengusulkan metode *clustering* untuk pengelompokan mahasiswa berdasarkan kemampuan penguasaan teknologi informasi, sehingga mahasiswa dapat dikelompokan sesuai dengan kemiripan/kedekatan tingkat kemampuan penguasaan ICT yang dimilikinya.

Clustering atau analisis *cluster* merupakan proses tak terawasi yang mengelompokan data yang identik ke dalam yang sama (Behera et al., 2021). *Clustering* memisahkan sekumpulan variabel hasil pengukuran atau perhitungan ke dalam kelompok-kelompok yang homogen, di mana anggota pada setiap kelompok tersebut memiliki kemiripan (Novoselsky & Kagan, 2021). *Clustering* adalah sebuah konsep untuk menentukan pola melalui pemetaan dan analisis data (Velmurugan, 2018). *Clustering* mempelajari kesamaan dari beberapa sampel data, dengan mengelompokkan data ke dalam kelas-kelas atau *cluster* sedemikian rupa sehingga objek-objek dalam suatu *cluster* memiliki kemiripan yang tinggi. *Clustering* mengelompokkan data atau objek menjadi kelompok-kelompok yang serupa berdasarkan karakteristik atau atribut yang dimiliki oleh objek tersebut. Pengelompokan tersebut mengacu pada pemecahan sekumpulan data menjadi kelompok-kelompok menurut kriteria yang sesuai dengan mengasosiasikan sampel data melalui kedekatan, kemiripan, atau ketidakmiripan (Mcilhany & Wiggins, 2018). Tujuannya adalah untuk mengidentifikasi pola alami atau struktur dalam data sehingga data yang serupa dikelompokkan bersama dalam satu kelompok. Dalam *clustering* dikelanal istilah *distance* atau *disimilarity* dan *similarity.* Keduanya merupakan dasar dalam mengembangkan algoritma *clustering* yang menggambarkan sejauh mana dua objek atau data memiliki kemiripan (*similarity*) satu sama lain berdasarkan atribut-atribut yang dimiliki objek tersebut. *Distance* biasanya digunakan pada fitur data kuantitatif, sedangkan *similarity* digunakan jika berhadapan dengan fitur data kualitatif (Xu & Tian, 2015).

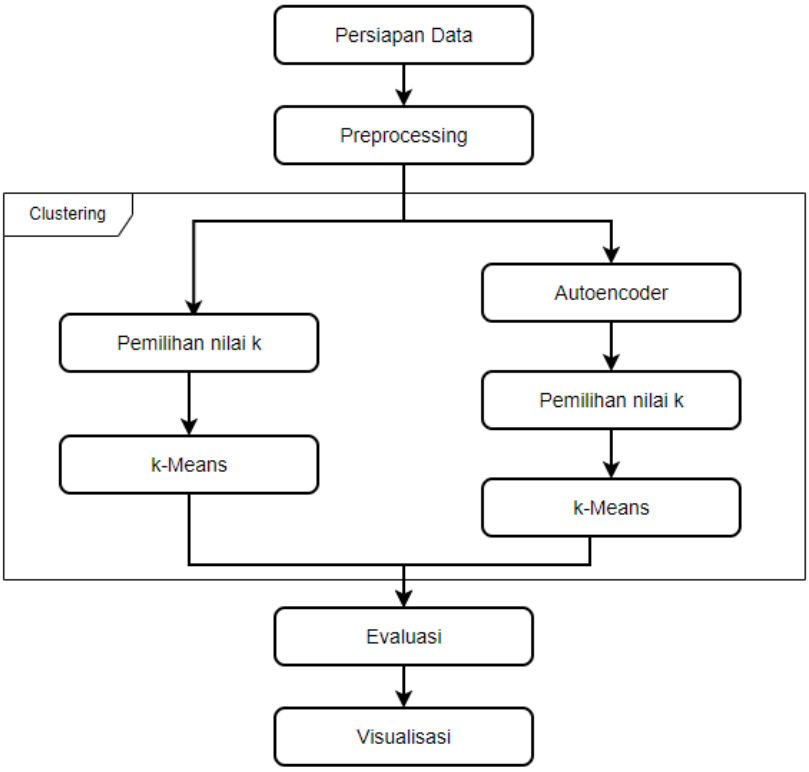
Salah satu algoritma *clustering* yang terkenal adalah k-Means. Secara teoritis k-Means merupakan metode yang sederhana, cepat menyatu, dan dapat secara efektif menangani kumpulan data berukuran kecil dan menengah (Zhao & Zhou, 2021). Algoritma k-Means relatif sederhana untuk diimplementasikan dan dipahami. K-Means mengelompokkan objek-objek di dalam *dataset* ke dalam kelompok atau *cluster* berdasarkan kemiripan atau jarak antara objek tersebut. K-means cocok untuk kumpulan data dengan jumlah data yang besar dan dimensi fitur yang tinggi, serta ketergantungannya pada data yang rendah (Wu et al., 2021). Algoritma ini mengelompokkan data menjadi k *cluster*, di mana k adalah jumlah cluster yang ditentukan sebelumnya oleh pengguna. Hasil dari k-Means adalah pengelompokan titik data ke dalam *cluster* yang jelas dan mudah diinterpretasikan. Ini membuatnya berguna untuk analisis eksploratif dan pemahaman pola dalam data.

K-Means telah banyak digunakan untuk pengelompokan data di berbagai bidang. Di bidang kesehatan, k-Means digunakan untuk mengelompokan balita berdasarkan kecukupan gizi (Dona & Rifqi, 2022). Pada penelitian tersebut, status gizi pada balita dikelompokan ke dalam dua *cluster* dengan tujuan untuk membantu pemantauan kebutuhan gizi pada balita di Kabupaten Rokan Hulu. Penelitian selanjutnya adalah penggunaan analisis *cluster* untuk mengelompokan kabupaten/kota di Provinsi Kalimantan Barat berdasarkan data pengguna alat kontrasepsi menggunakan algoritma k-Means dan k-Medoids (Musfiani, 2019). Di sektor perkebunan, *analisis*

cluster digunakan untuk pengelompokan daerah produksi kakao di Provinsi Sulawesi Selatan (Abidin et al., 2022). Penelitian tersebut membandingkan algoritma k-Means dan k-Medoids berdasarkan nilai *Davies-Bouldin Index (DBI)* pada *RapidMiner* dan disimpulkan bahwa algoritma k-Means lebih efektif dibandingkan dengan k-Medoids. Di sektor perbankan, algoritma k-Means lebih efektif dan efisien digunakan untuk pengelompokan kredit macet (Fitriani et al., 2023). Di bidang pendidikan, algoritma k-Means digunakan dalam penelitian tersebut untuk mengelompokan siswa berdasarkan prestasi, sehingga dapat ditentukan siswa mana yang dapat masuk ke dalam kelas unggulan (Nur Aziz & Zuliarso, 2022).

Autoencoder adalah jenis arsitektur jaringan saraf tiruan (*neural network*) yang digunakan dalam pembelajaran tak terawasi (*unsupervised learning*). *Autoencoder* dirancang untuk menghasilkan representasi data yang lebih ringkas dan informasi yang lebih penting dengan cara mengompresi dan kemudian mendekompresi data masukan. *Autoencoder* sering digunakan untuk ekstraksi fitur, pengurangan dimensi, *denoising* data, dan berbagai aplikasi dalam analisis data dan pembelajaran mesin. Seiring dengan perkembangan pembelajaran mendalam (*deep learning*), *autoencoder* menjadi yang terdepan dalam pemodelan generatif (Zhai et al., 2018). *Autoencoder* terdiri dari sepasang dua jaringan yang terhubung: *encoder* dan *decoder* (Nugroho et al., 2020). Bagian pertama dari *autoencoder*, yang disebut *encoder*, yaitu lapisan-lapisan yang bertanggung jawab untuk mengambil data masukan dan mengubahnya menjadi representasi yang lebih ringkas. Bagian kedua dari *autoencoder* disebut *decoder.* Bagian ini berisi lapisan-lapisan bertanggung jawab untuk mengambil representasi terkompresi (kode) dari *encoder* dan mendekompresinya untuk menghasilkan rekonstruksi data yang serupa dengan data masukan. Di antara kedua bagian tersebut terdapat sebuah lapisan yang disebut dengan *bottleneck layer,* yaitu lapisan yang berfungsi sebagai representasi terkompresi atau kode dari data. Prinsip utama dari *autoencoder* adalah untuk meminimalkan perbedaan antara data masukan dan rekonstruksi *output* yang dihasilkan oleh *decoder*. Dalam proses pelatihan, *autoencoder* berusaha untuk belajar merepresentasikan data menjadi lebih baik dan lebih ringkas.

2. METODE PENELITIAN



Gambar 1 Tahapan dalam Analisis *Cluster*

Penelitian dimulai dengan persiapan data, termasuk di dalamnya pengumpulan data dan analisis data. Kemudian dianjutkan dengan *preprocessiong* data dengan melakukan pembersihan data dan normalisasi atau penskalaan data. Selanjutnya data akan digunakan dalam proses *clustering*. Proses *clustering* dilakukan dengan menggunakan dua metode, yaitu k-Means dan k-Means + *autoencoder*. Hasil dari proses *clustering* kemudian dibandingkan dan dievaluasi. Tahapan dalam analisis *cluster* menggunakan k-Means dan *autoencoder* dapat dilihat pada Gambar 1.

2.1 Data dan Sumber Data

Pengumpulan data menjadi langkah awal dalam penelitian, terutama untuk penelitian yang terkait dengan analisis data. Data merupakan sekumpulan fakta yang dapat diamati, diukur, dan diolah. Dalam penelitian ini, data yang digunakan adalah data nilai pretest kemampuan mahasiswa UIN Sunan Kalijaga angkatan tahun 2022. Pengambilan data dilakukan dengan meminta mahasiswa untuk mengerjakan soal pengetahuan dasar ICT yang terdiri dari empat bagian, yaitu tentang Microsoft Word, Microsoft Excel, Microsoft Power Point, dan Internet. Dari hasil *pretest* yang dilakukan diperoleh data sebanyak 3.382. Sampel data *pretest* dapat dilihat pada Tabel 1.

Tabel 1 Sampel Data Nilai *Pretest* ICT

Id	nim	nama	word	excel	power point	internet
1	2210*****01	Muh**********udin	56	36	52	56
2	2210*****07	Rion**********cky	76	44	68	68
3	2210*****09	Arfa******qi	64	72	60	52
4	2210*****37	Imel*********lfa	60	40	52	68
5	2210*****49	Muti**********hra	68	32	52	44
...	...	...	...	...	...	...
3378	2210*****76	Muh*********ndy	48	28	16	40
3379	2210*****79	Nail******ah	88	88	52	76
3380	2210*****94	Pris**********mah	48	24	60	48
3381	2210*****27	Ram******n	76	48	48	68
3382	2210*****89	Muh**********hul	84	64	48	40

2.2 *Pre-processing*

Pre-processing digunakan untuk memastikan bahwa data sudah bersih dan siap digunakan untuk analisis atau pemodelan. *Pre-processing* merupakan proses mengubah data mentah menjadi format yang lebih dimengerti (Agarwal, 2015). Tahap *pre-processing* data pada penelitian ini antara lain pembersihan data dan normalisasi data. Data yang tidak lengkap dapat mengganggu dapat mempengaruhi kualitas analisis dan model yang dibangun. Pembersihan data bertujuan untuk mengidentifikasi, menangani, dan mengatasi masalah di dalam data tersebut. Data yang bersih dan terstruktur dengan baik tentunya akan menghasilkan hasil analisis yang lebih akurat dan model yang lebih andal.

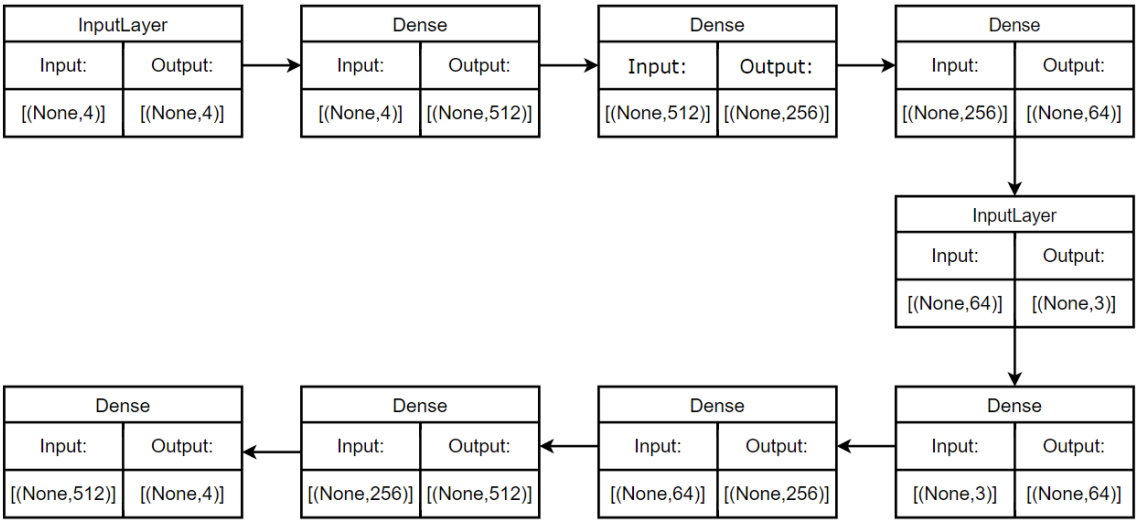
Normalisasi data bertujuan untuk mengubah skala data sehingga memiliki rentang nilai yang seragam. Perbedaan skala antar atribut sangat berpengaruh dalam analisis dan pemodelan pembelajaran mesin. Atribut dengan skala yang lebih besar akan dominan dalam analisis dan pemodelan. Oleh karena itu, skala atribut data harus diseragamkan sebelum digunakan dalam analisis atau pemodelan. Metode normalisasi yang sering digunakan antara lain *min-max normalization*, *robust scalling*, dan *Z-score normalization*. *Z-score normalization*, juga dikenal sebagai *standardization*, yaitu salah satu metode normalisasi yang umum digunakan dalam analisis data dan statistik. *Z-score* dihitung menggunakan nilai rata-rata dan *standar deviasi* (Prihanditya & Alamsyah, 2020). Tujuan utama dari *Z-score normalization* adalah mengubah skala data sehingga memiliki rata-rata (*mean*) bernilai nol dan simpangan baku (*standard deviation*) satu. Dengan kata lain, data yang telah dinormalisasi dengan *Z-score* memiliki distribusi yang berpusat di sekitar nol dan memiliki deviasi standar yang seragam. *Z-score normalization* diformulasikan pada Pers. (1). Di mana $X_{normalized}$ adalah nilai ternormalisasi, X

adalah nilai asli dalam *dataset*, μ adalah rata-rata di dalam *dataset*, dan σ adalah simpangan baku di dalam *dataset*.

$$X_{normalized} = \frac{X - \mu}{\sigma} \quad (1)$$

2.3 Pemodelan ANN Autoencoder

Autoencoder digunakan untuk ekstraksi fitur dan mereduksi dimensi, sehingga proses *clustering* menjadi lebih cepat. Seperti dijelaskan sebelumnya bahwa *dataset* yang digunakan memiliki dimensi 3.382 x 4, sehingga bagian *encoder* akan menerima input data dengan dimensi 4 pada lapisan input dan diikuti oleh tiga lapisan *dense* dengan jumlah neuron masing-masing 512, 256, dan 64. Bagian *decoder* terdiri dari empat lapisan dengan jumlah neuron masing-masing 64, 256, 512, dan 4. Di antara bagian *encoder* dan *decoder* terdapat lapisan *code* dengan jumlah neuron sebanyak 3. Arsitektur model *autoencoder* dapat dilihat pada Gambar 2.



Gambar 2 Arsitektur Model *Autoencoder*

2.4 Pemilihan Nilai k

K-Means tidak dapat mengetahui jumlah *cluster* yang optimal. Metode *Elbow* dapat digunakan untuk untuk mengoptimalkan jumlah *cluster* pada metode k-Means (Marutho et al., 2018). Langkah yang harus dilakukan dalam menentukan nilai k menggunakan metode *Elbow* adalah menghitung *Sum of Square Error (SSE)* seperti pada Pers. (2) untuk setiap nilai yang akan menjadi kandidat nilai k. SSE menggambarkan seberapa jauh objek-objek di dalam sebuah cluster dari pusatnya. Kemudian SSE pada setiap k digambarkan dalam bentuk grafik sehingga membentuk siku pada grafik tersebut. Jumlah *cluster* atau nilai k terbaik akan diambil dari nilai SSE yang mengalami penurunan dan berbentuk siku (Nainggolan et al., 2019). Pada perhitungan SSE *n* merupakan jumlah data, k merupakan jumlah *cluster*, x_{ij} adalah data ke-i di dalam *cluster* ke-j, dan c_j merupakan pusat *cluster* ke-j.

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} (x_{ij} - c_j)^2 \quad (2)$$

2.5 *Clustering* dengan K-Means

K-Means bertujuan untuk meminimalkan jumlah jarak antara objek-objek di dalam *cluster* dengan pusat *cluster* masing-masing, dan mengelompokkannya secara iteratif (Cui, 2020). Algoritma ini

mengelompokkan data menjadi k *cluster*, di mana k adalah jumlah *cluster* yang ditentukan sebelumnya. Langkah-langkah umum dalam algoritma k-Means adalah sebagai berikut:

1) Pilih titik awal yang akan berperan sebagai pusat *cluster* awal (*centroid*). Inisialisasi pusat *cluster* ini dapat dilakukan secara acak atau berdasarkan metode khusus.
2) Hitung jarak objek data observasi dengan setiap pusat *cluster*. Masukan data tersebut ke dalam *cluster* berdasarkan jarak terdekat dengan pusat *cluster*.
3) Tentukan pusat *cluster* baru untuk setiap cluster dengan menghitung rata-rata jarak antara pusat *cluster* dengan semua data di dalam *cluster* tersebut.
4) Ulangi langkah 2 dan 3 hingga tidak ada lagi data yang berpindah *cluster* atau hingga batasan iterasi tertentu tercapai.

2.6 Evaluasi Model Clustering

Evaluasi model dilakukan untuk mengukur kinerja model *clustering*. Metode yang sering digunakan untuk mengevaluasi model *clustering* adalah *Silhouette coefficient* dan *Davis-Bouldin Index (DBI)*. *Silhouette coefficient* menggambarkan seberapa dekat suatu data dengan data lainnya di dalam *cluster* yang sama dibandingkan dengan data pada cluster lainnya. *Silhouette coefficient* memiliki rentang nilai antara -1 sampai 1. Nilai *Silhouette* yang mendekati 1 menunjukkan adanya hubungan yang erat antara objek dengan cluster di mana objek tersebut berada (Yuan & Yang, 2019). *Silhouette coefficient* dinotasikan dengan *S(i)* untuk setiap data i yang dihitung melalui Pers. (3). Di mana a(i) merupakan rata-rata jarak suatu objek data dengan objek data lainnya di dalam *cluster* yang sama, b(i) adalah rata-rata jarak suatu objek data dengan objek data pada *cluster* yang lain.

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} = \begin{cases} 1 - \dfrac{a(i)}{b(i)}, & a(i) > b(i) \\ 0, & a(i) = b(i) \\ \dfrac{b(i)}{a(i)} - 1, & a(i) < b(i) \end{cases} \quad (3)$$

Davis-Bouldin Index (DBI) adalah suatu metrik evaluasi untuk mengukur kualitas model *clustering*. DBI mengukur seberapa jauh suatu *cluster* dari *cluster* lain dalam hal varian data dan jarak antar-pusat *cluster*. Jumlah *cluster* yang memiliki DBI yang minimal dianggap sebagai jumlah *cluster* yang optimal (Baser & R. Saini, 2015). Nilai DBI yang semakin rendah menunjukan *cluster* yang terbentuk semakin baik. Nilai DBI dihitung sebagai ukuran rata-rata kemiripan setiap *cluster* dengan *cluster* lain yang paling mirip dengannya. Dalam hal ini, kemiripan diartikan sebagai perbandingan antara jarak antar *cluster* dan jarak *intra-cluster*. Nilai DBI untuk jumlah k *cluster* dapat dihitung dengan Pers. (4). Di mana $\Delta(x_k)$ merupakan dispersi atau jarak sebaran data di dalam *cluster* k dan $\partial(x_i, x_j)$ adalah jarak pusat *cluster* i dengan pusat *cluster* j.

$$DBI = \frac{1}{k}\sum_{i=i}^{k} \max\left(\frac{\Delta(x_i) + \Delta(j)}{\partial(x_i, x_j)}\right. \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 Persiapan Data

Persiapan data dilakukan untuk memastikan data yang digunakan sesuai dengan tujuan pemodelan. Analisis awal dilakukan untuk memahami struktur, distribusi, dan karakteristiknya, serta mengidentifikasi variabel-variabel beserta pemahaman makna masing-masing variabel. Data *pretest* ICT memiliki empat variabel utama yang akan digunakan dalam analisis *cluster*, yaitu nilai word, excel, power point, dan internet. Deskripsi variabel data *pretest* ICT disajikan pada Tabel 2. Variabel word memiliki rata-rata yang lebih tinggi dibandingkan variabel lainnya,

sedangkan variabel excel memiliki rata-rata yang paling rendah. Standar deviasi variabel internet lebih kecil dibanding lainnya, artinya sebaran data cenderung berkumpul mendekati rata-rata.

Tabel 2 Deskripsi Data

	word	excel	power point	internet
count	3382	3382	3382	3382
mean	64.9533	41.0089	48.7912	58.5795
std	16.1293	16.2034	14.4353	12.0773
min	4	0	4	12
25%	52	28	40	52
50%	68	40	48	60
75%	76	52	60	68
max	100	88	96	96

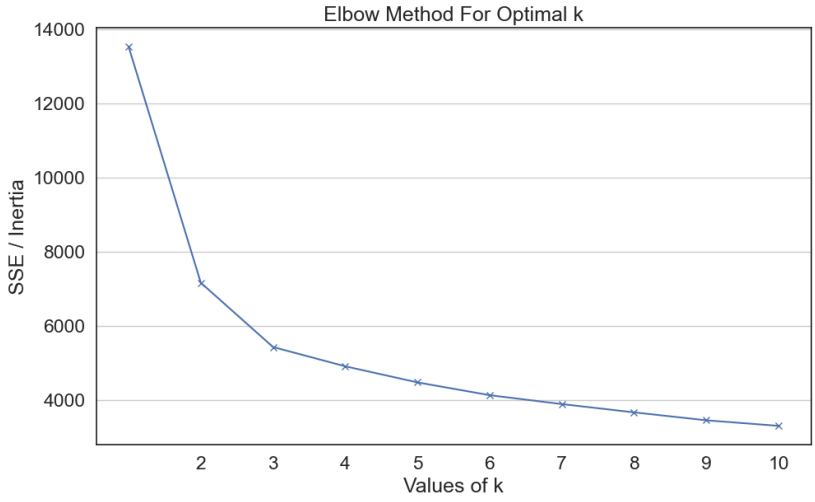
3.2 Normalisasi Data

Sebelum *dataset* digunakan dalam pemodelan *clustering*, *dataset* harus distandarkan agar semua variabel memiliki rentang data yang sama, sehingga tidak ada variabel yang lebih atau kurang dominan. Normalisasi data dilakukan menggunakan metode Z-Score *normalization*, sehingga semua variabel memiliki rata-rata 0 dan simpangan baku 1. Hasil normalisasi data disajikan pada Tabel 3.

Tabel 3 Sampel Data Nilai *Pretest* ICT

	Word	Excel	Power Point	Internet
0	-0.55518	-0.30917	0.222318	-0.21362
1	0.684986	0.184627	1.330879	0.780126
2	-0.05911	1.912919	0.776599	-0.54486
3	-0.30714	-0.06227	0.222318	0.780126
4	0.188921	-0.55607	0.222318	-1.20736
...	...	...	...	...
3377	-1.05124	-0.80297	-2.27194	-1.53861
3378	1.429083	2.900514	0.222318	1.442622
3379	-1.05124	-1.04987	0.776599	-0.87611
3380	0.684986	0.431526	-0.05482	0.780126
3381	1.181051	1.419121	-0.05482	-1.53861

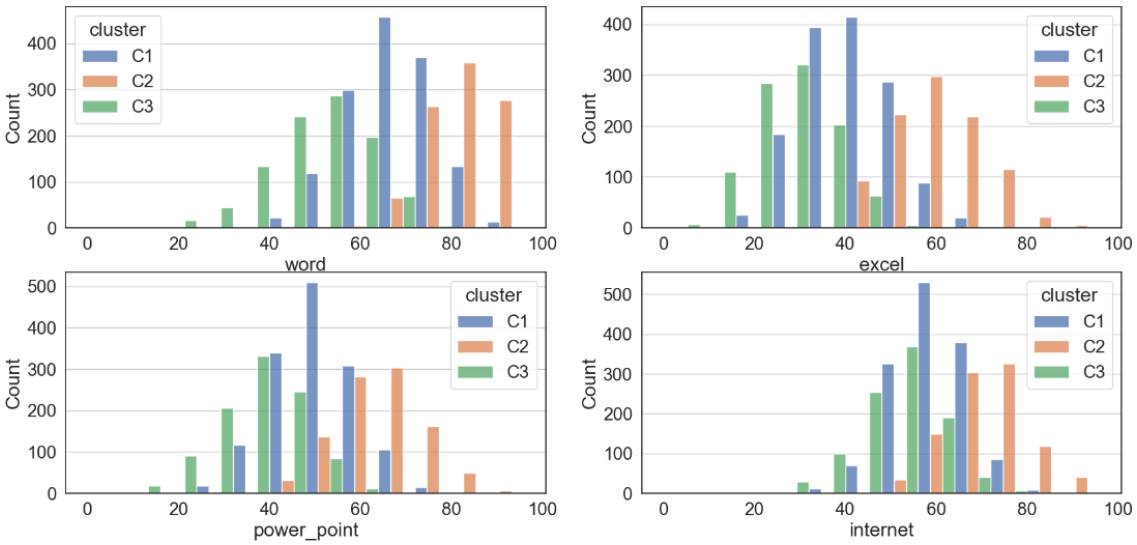
3.3 Proses *Clustering* dengan K-Means



Gambar 3 Grafik Inertia untuk Setiap k

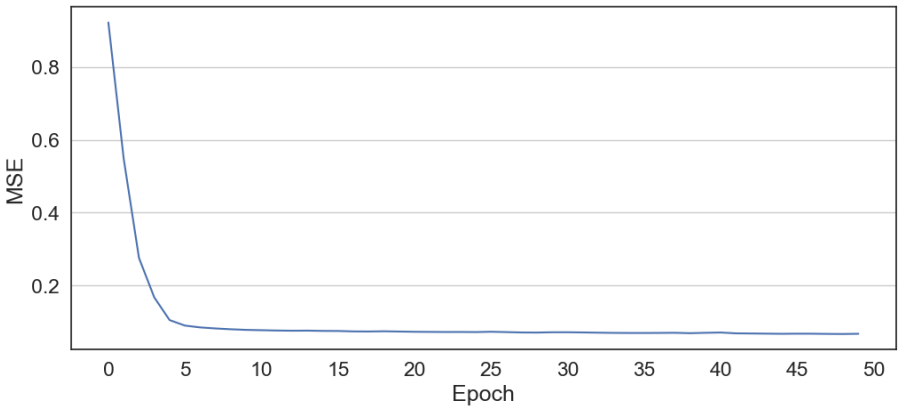
Data hasil normalisasi selanjutnya dapat digunakan dalam proses *clustering*. Dalam proses *clustering*, nilai k atau jumlah *cluster* harus ditentukan sebelumnya. Untuk menentukan jumlah *cluster* yang optimal, metode yang digunakan adalah metode *Elbow*. Metode tersebut menghitung nilai SSE untuk setiap nilai k yang dicoba. Grafik SSE untuk setiap nilai k dapat dilihat pada Gambar 3.

Dari Gambar 4 terlihat bahwa grafik SSE membentuk siku pada k=3, sehingga jumlah *cluster* yang digunakan adalah tiga. Setelah jumlah *cluster* ditentukan, selanjutnya proses *clustering* dapat dilakukan menggunakan *dataset* yang sudah dinormalisasi. Proses *clustering* dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan fungsi *KMeans* yang disediakan oleh pustaka *scikit-learn*. *Scikit-learn* merupakan sebuah pustaka (*library*) dalam bahasa pemrograman Python untuk pembelajaran mesin (*machine learning*) dan tugas-tugas analisis data. Dari proses *clustering* diperoleh tiga cluster dengan jumlah anggota pada *cluster* pertama (C1) sebanyak 993 anggota, *cluster* kedua (C2) sebanyak 1.414 anggota, dan *cluster* ketiga (C3) sebanyak 975 anggota. Sebaran data untuk setiap *cluster* hasil proses *clustering* dengan k-Means dapat dilihat pada Gambar 4.



Gambar 4 Sebaran Data Hasil Proses *Clustering* dengan K-Means

3.4 Pemodelaan *Autoencoder*



Gambar 5 Grafik MSE Proses Pelatihan Model *Autoencoder*

Pemodelan *autoencoder* dilakukan menggunakan *keras* dari *Tensorflow* yang dapat mempermudah dalam mendifinisikan model jaringan syaraf tiruan dan melatih model tersebut di

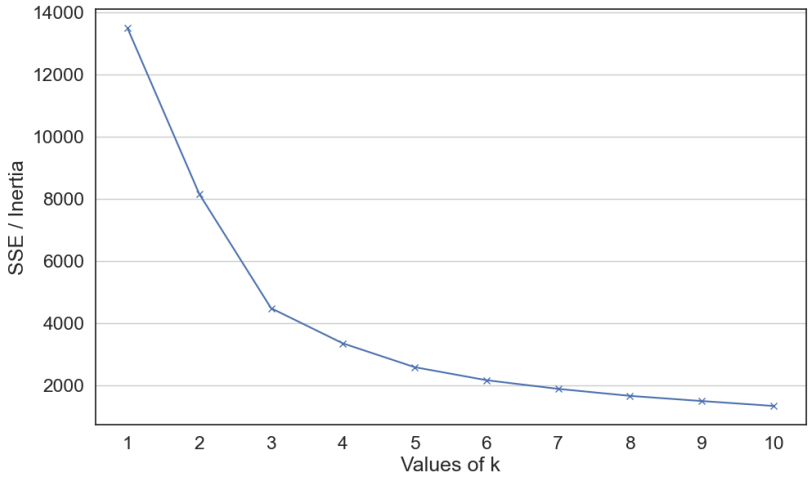
dalam bahasa pemrograman Python. Proses pelatihan model dilakukan sebanyak 50 *epoch* dengan menggunakan fungsi optimasi *Adam (Adaptive Moment Estimation)* dan *loss function MSE (Mean Squared Error)*. Grafik MSE yang dihasilkan dapat dilihat pada Gambar 5. Langkah selanjutnya adalah merekonstruksi *dataset* dengan model *autoencoder*. Hasil rekonstruksi *dataset* disajikan pada Tabel 4.

Tabel 4 Hasil Rekontruksi *Dataset* Menggunakan Model *Autoencoder*

	X1	X2	X3
0	2.055754	1.752693	1.488532
1	2.220366	3.880299	1.876992
2	1.101668	3.291011	3.203516
3	1.769379	1.634185	0.768939
4	2.355839	2.599715	2.975083
...	...	...	...
3377	2.95972	0.899038	3.391938
3378	0.272069	3.586121	1.772966
3379	3.777638	3.445621	2.890459
3380	0.727395	1.381394	0.600213
3381	0.904083	2.745968	3.974567

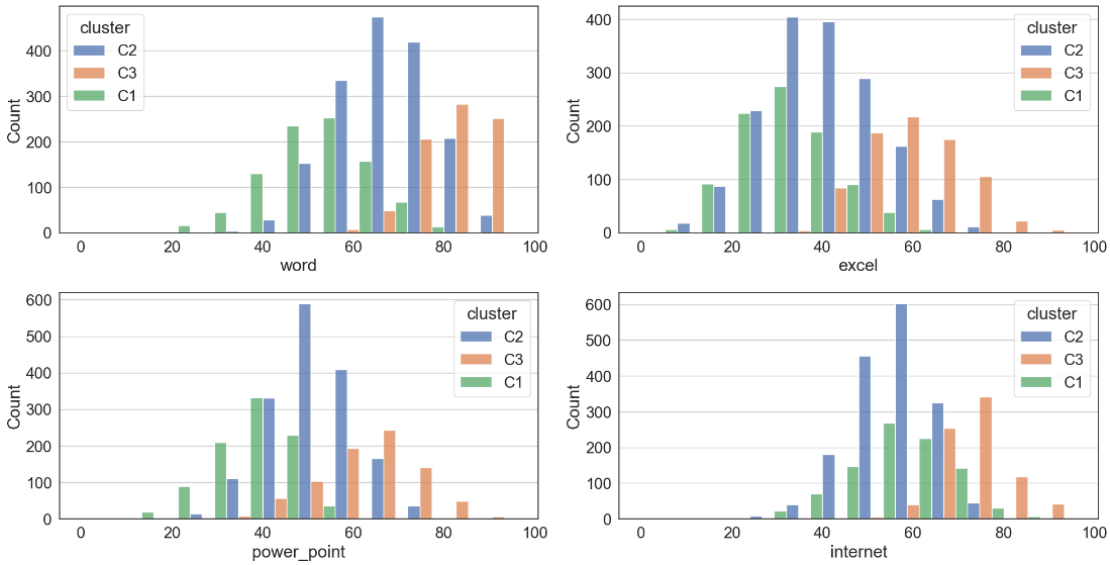
3.5 *Clustering* dengan Kombinasi K-Means dan *Autoencoder*

Data hasil transformasi menggunakan model *autoencoder* selanjutnya diproses menggunakan k-Means. Pada proses *clustering* yang kedua, *dataset* hanya memiliki tiga variabel. Seperti dalam proses *clustering* sebelumnya, nilai k harus ditentukan dengan mencoba satu persatu sehingga diperoleh nilai k yang optimal. Pemilihan nilai k di dalam proses *clustering* ini juga dilakukan dengan metode *Elbow*. Grafik SSE untuk menentukan nilai k yang optimal pada proses *clustering* menggunakan k-Means dan *autoencoder* dapat dilihat pada Gambar 6.



Gambar 6 Grafik SSE untuk Proses *Clustering* K-Meas dan *Autoencoder*

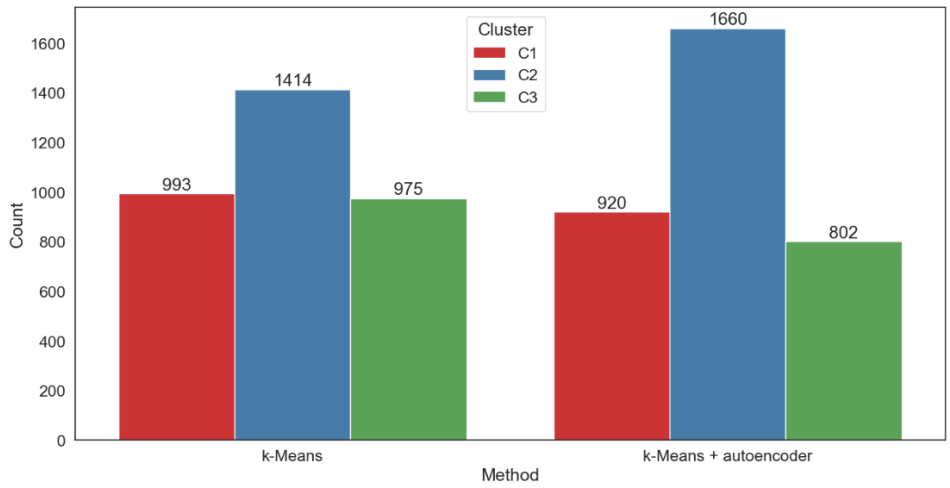
Dari Gambar 6 terlihat bahwa grafik SSE membentuk siku pada k=3, sehingga nilai k yang digunakan dalam clustering juga 3. *Cluster* C1 dengan jumlah anggota sebanyak 920 diisi oleh data yang memiliki nilai rendah untuk semua variabel. *Cluster* C2 dengan jumlah anggota sebanyak 1.660 diisi oleh data dengan nilai sedang untuk setiap variabelnya. Sedangkan *cluster* C3 dengan jumlah anggota sebanyak 802 berisi data yang memiliki nilai setiap variabel lebih tinggi dibandingkan pada *cluster* lainnya. Sebaran data berdasarkan hasil *clustering* menggunakan k-Means dan *autoencoder* dapat dilihat pada Gambar 7.



Gambar 7 Sebaran Data Hasil *Clustering* dengan K-Means dan *Autoencoder*

3.6 Evaluasi Model *Clustering*

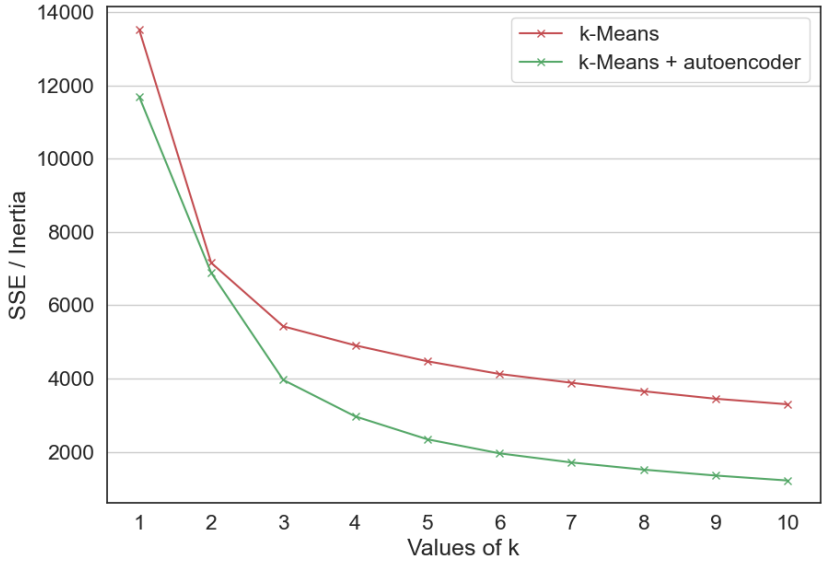
Dari kedua proses *clustering* menggunakan k-Means dan kombinasi k-Means + *autoencoder*, keduanya menghasilkan tiga *cluster* yang dianggap optimal. Perbandingan hasil *clustering* keduannya berdasarkan jumlah anggota setiap *cluster* disajikan pada Gambar 8. Pada proses *clustering* menggunakan kombinasi k-Means dan *autoencoder* menghasilkan nilai SSE yang lebih rendah dibandingkan hanya dengan k-Means, artinya jarak antara titik-titik dengan pusat *cluster* untuk setiap *cluster* pada k-Means + *autoencoder* juga lebih dekat. Pada nilai k=3, *clustering* menggunakan k-Means saja menghasilkan inertia sebesar 5.424,77, sedangkan *clustering* dengan kombinasi k-Means dan *autoencoder* menghaslakan SSE sebesar 2.966,64. Perbandingan SSE antara k-Means dan kombinasi k-Means + *autoencoder* dapat dilihat pada Gambar 9.



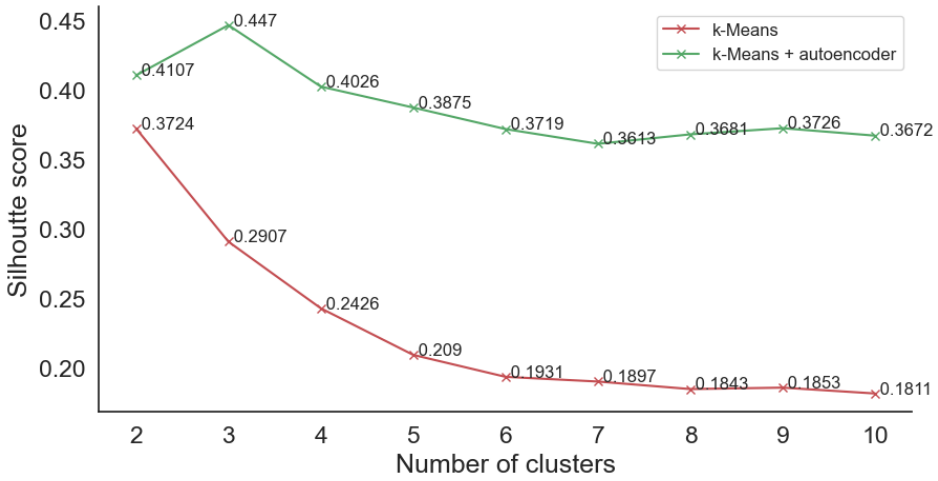
Gambar 8 Perbandingan Hasil *Clustering* Berdasarkan Jumlah Setiap *Cluster*

Validasi selanjutnya dilakukan dengan *Silhouette coefficient* dan *Davis-Bouldin Index (DBI)*. Skor *silhouette* pada k=3 yang diperloleh model *clustering* dengan kombinasi k-Means dan *autoencoder* lebih besar dibandingkan hanya dengan k-Means saja, yaitu 0,447 berbanding 0,2907. Pada *clustering* tanpa *autoencoder*, nilai *silhouette* terus menurun ketika diujikan untuk

setiap nilai k = 2 hingga k=10, dan cenderung tidak menemukan nilai k yang optimal di atas k=2. Perbandingan skor *silhouette* yang diperoleh dari proses *clustering* menggunakan k-Means tanpa *autoencoder* dan dengan *auoencoder* dapat dilihat pada Gambar 10.



Gambar 9 Perbandingan SSE untuk Nilai k=1 Sampai k=10

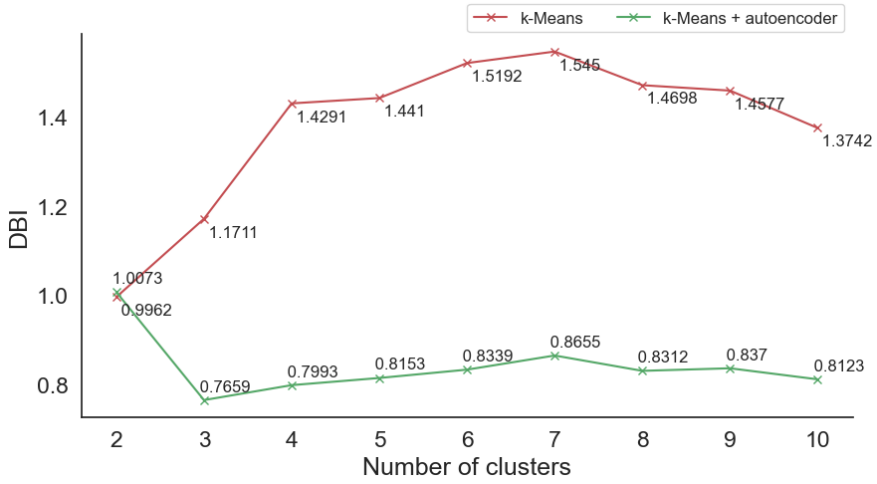


Gambar 10 Perbandingan Nilai Silhoutte

Dari hasil evaluasi model menggunakan *Davis-Bouldin Index (DBI)* diketahui bahwa model *clustering* kombinasi k-Means dan *autoencoder* lebih kecil dibandingkan hanya menggunakan k-Means saja. Pada nilai k=3, nilai DBI yang dihasilkan sebesar 0,7658, sedangkan *clustering* tanpa *autoencoder* menghasilkan nilai DBI sebesar 1.1711. Jika semua kandidat jumlah *cluster* k=2 sampai k=10 divalidasi menggunakan DBI, model *clustering* dengan kombinasi k-Means dan *autoencoder* juga lebih baik dibandingkan tanpa *autoencoder*. Perbandingan nilai DBI pada model *clustering* menggunakan k-Means tanpa *autoencoder* dan dengan *autoencoder* dapat dilihat pada Gambar 11.

Dari hasil evaluasi, diketahui bahwa kombinasi k-Means dan *autoencoder* menghasilkan kinerja yang lebih baik dibandingkan hanya dengan menggunakan k-Means saja, yang ditunjukan dengan nilai *Silhouette* yang lebih tinggi dan nilai DBI yang lebih rendah. Oleh karena itu,

penggunaan *autoencoder* untuk ekstraksi fitur dan reduksi dimensi dapat meningkatkan kinerja model *clustering,* khususnya *clustering* menggunakan algoritma k-Means.



Gambar 11 Perbandingan Indeks Davies-Bouldin

4. KESIMPULAN

Kombinasi k-Means dan *autoencoder* dapat digunakan dengan cukup baik untuk pengelompokan kemampuan penguasaan ICT mahasiswa. Hasil pengukuran dan evaluasi SSE, *Silhouette Cofficient*, dan *Davies-Bouldin Index* menunjukan bahwa penggabungan k-Means dengan *autoencoder* dapat meningkatkan kinerja model *clustering* pada data kemampuan ICT mahasiswa dibandingkan hanya menggunakan k-Means saja. Hal ini ditunjukan dengan nilai SSE yang lebih rendah, nilai koefisien *Silhouette* yang lebih tinggi, dan niilai DBI yang lebih rendah. Meskipun demikian, masih terdapat beberapa data yang tumpang tindih. Oleh karena itu, dalam penelitian selanjutnya dapat melakukan analisis *cluster* pada data kemampuan ICT mahasiswa menggunakan model baru atau kombinasi model-model yang sudah ada.

UCAPAN TERIMA KASIH

Ucapan terima kasih dihaturkan kepada LPPM UIN Sunan Kalijaga Yogyakarta dan Pusat teknologi Informasi dan Pangkalan Data atas segala dukungan dan fasilitas yang diberikan. Penelitian ini didasarkan pada Penelitian BOPTN Tahun 2023.

DAFTAR PUSTAKA

Abidin, N. A. S. Z., Avila, R. D., Hermatyar, A., & Rismayani, R. (2022). Perbandingan Algoritma K-Means dan K-Medoids untuk Pengelompokan Daerah Produksi Kakao. *Jurnal Teknik Informatika dan Sistem Informasi*, *8*(2). https://doi.org/10.28932/jutisi.v8i2.4897

Agarwal, V. (2015). Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis. *International Journal of Computer Applications*, *131*(4), 30–36. https://doi.org/10.5120/ijca2015907309

Baser, P., & R. Saini, J. (2015). Agent based Stock Clustering for Efficient Portfolio Management. *International Journal of Computer Applications*, *116*(3), 35–41. https://doi.org/10.5120/20317-2381

Behera, M. P., Sarangi, A., & Mishra, D. (2021). K-medoids Crazy Firefly Algorithm for Unsupervised Data Clustering. *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)*, 1–6. https://doi.org/10.1109/ODICON50556.2021.9428980

Cui, M. (2020). Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. *Accounting, Auditing and Finance Clausius Scientific Press*, *1*(1). https://doi.org/10.23977/accaf.2020.010102

Dona, D., & Rifqi, M. (2022). Penerapan Metode K-Means Clustering untuk Menentukan Status Gizi Baik dan Gizi Buruk pada Balita (Studi Kasus Kabupaten Rokan Hulu). *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, *7*(2), 179–191. https://doi.org/10.36341/rabit.v7i2.2171

Fitriani, M. N. R., Priyatna, B., Huda, B., Hananto, A. L., & Tukino, T. (2023). Implementasi Metode K-Means untuk Memprediksi Status Kredit Macet. *Jurnal Sistem Komputer dan Informatika (JSON)*, *4*(3), 554. https://doi.org/10.30865/json.v4i3.5953

Marutho, D., Handaka, S. H., Wijaya, E., & Muljono, M. (2018). The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News. *2018 International Seminar on Application for Technology of Information and Communication*, 533–538. https://doi.org/10.1109/ISEMANTIC.2018.8549751

Mcilhany, K., & Wiggins, S. (2018). High Dimensional Cluster Analysis Using Path Lengths. *Journal of Data Analysis and Information Processing*, *06*(03), 93–125. https://doi.org/10.4236/jdaip.2018.63007

Musfiani, M. (2019). Analisis Cluster dengan Menggunakan Metode Partisi pada Pengguna Alat Kontrasepsi di Kalimantan Barat. *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, *8*(4), 893–902. https://doi.org/10.26418/bbimst.v8i4.36584

Nainggolan, R., Perangin-angin, R., Simarmata, E., & Tarigan, A. F. (2019). Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) Optimized by Using the Elbow Method. *Journal of Physics: Conference Series*, *1361*(1), 012015. https://doi.org/10.1088/1742-6596/1361/1/012015

Novoselsky, A., & Kagan, E. (2021). An Introduction to Cluster Analysis. In *An Introduction to Toxicogenomics* (pp. 1–9). Chapman and Hall/CRC. https://doi.org/10.13140/RG.2.2.25993.57448/1

Nugroho, H., Susanty, M., Irawan, A., Koyimatu, M., & Yunita, A. (2020). Fully Convolutional Variational Autoencoder for Feature Extraction of Fire Detection System. *Jurnal Ilmu Komputer dan Informasi*, *13*(1), 9–15. https://doi.org/10.21609/jiki.v13i1.761

Nur Aziz, Y. A., & Zuliarso, E. (2022). Sistem Penerimaan Siswa Baru di Smkn 3 Pati Berdasar Jalur Prestasi Menggunakan Algoritma Klastering K-Means Berbasis Web. *Jurnal Ilmiah Informatika*, *10*(02), 86–95. https://doi.org/10.33884/jif.v10i02.5555

Prihanditya, H. A., & Alamsyah. (2020). The Implementation of Z-Score Normalization and Boosting Techniques to Increase Accuracy of C4.5 Algorithm in Diagnosing Chronic Kidney Disease. *Journal of Soft Computing Exploration*, *1*(1), 63–69. https://doi.org/10.52465/joscex.v1i1.8

Velmurugan, T. (2018). A State of Art Analysis of Telecommunication Data by k-Means and k-Medoids Clustering Algorithms. *Journal of Computer and Communications*, *06*(01), 190–202. https://doi.org/10.4236/jcc.2018.61019

Wu, C., Yan, B., Yu, R., Yu, B., Zhou, X., Yu, Y., & Chen, N. (2021). K-Means Clustering Algorithm and Its Simulation Based on Distributed Computing Platform. *Complexity*, *2021*(1). https://doi.org/10.1155/2021/9446653

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, *2*(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1

Yuan, C., & Yang, H. (2019). Research on K-Value Selection Method of K-Means Clustering Algorithm. *J*, *2*(2), 226–235. https://doi.org/10.3390/j2020016

Zhai, J., Zhang, S., Chen, J., & He, Q. (2018). Autoencoder and Its Various Variants. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 415–419. https://doi.org/10.1109/SMC.2018.00080

Zhao, Y., & Zhou, X. (2021). K-means Clustering Algorithm and Its Improvement Research. *Journal of Physics: Conference Series*, *1873*(1), 012074. https://doi.org/10.1088/1742-6596/1873/1/012074