Available online at https://ejournal.almaata.ac.id/index.php/IJUBI
Indonesian Journal of Business Intelligence
Volume 4 | Issue 2 | December (2021)
IJUBI
Indonesian Journal ---- of ---- Business Intelligence

PENGARUH REDUKSI DIMENSI TERHADAP METODE PENGKLASTERAN BERBASIS CENTROID DAN METODE PENGKLASTERAN BERBASIS DENSITY DALAM PENGKLASTERAN DOKUMEN TEKS

Muhammad Ihsan Jambak[1], Rusdi Efendi[2]
[1,2]Program Studi Manajemen Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya.
jambak@unsri.ac.id , rusdiefendi8@gmail.com
Jl. Sri Jaya Negara, Bukit Besar, Palembang 30139, Sumatera Selatan, Indonesia

Keywords:

Dimension Reduction, Clustering, k-Means, DBSCAN.

Abstract

Density-based clustering is usually more effective when processing data of different densities. This method is pioneered by the Density-based Applied Noise Spatial Clustering (DBSCAN) algorithm. There is a significant difference in behavior between k-Means and DBSCAN, which is processing data that contains noise. To this end, this research studies the impact of dimensionality reduction on high-dimensional data on the clustering results of the k-Means algorithm represented by the centroid method and the clustering results of the DBSCAN algorithm represented by the density method. Although the quality of the clustering results on k-Means has been improved after the numerical reduction by Singular Value Decomposition (SVD), from the initial average distance of 1.04136 to 0.003, the statistical change is not significant or considered to be the same. Therefore, it can be concluded statistically that SVD has no effect on the quality of k-Means clustering results. On the other hand, in DBSCAN, the effect of SVD dimensionality reduction is very significant. It can change the quality of the clustering results from the initial average intra-cluster distance of 76.13480 to 13.71130 or improve the quality by 555.27%. The significant impact of SVD on SVD + k-Means optimization and SVD + DBSCAN optimization cluster calculation time changes is also shown. SVD optimization can accelerate k- Means calculation time from 3.68182 seconds to 2,09091 seconds or 1.76 times. At the same time, SVD optimization accelerates the DBSCAN calculation time from 19.40000 seconds to 0.97500 seconds or 19.89 times.

Kata Kunci:

hingga 5 kata kunci, ditulis miring, dengan pemisah koma.

Abstrak

Pengklasteran berbasis kepadatan biasanya lebih efisien bekerja pada data dengan kepadatan yang berbeda, dimana metoda ini dipelopori oleh algoritma Density-Based Spatial Clustering of Application with Noise (DBSCAN). Terdapat perbedaan perilaku yang signifikan antara k-Means dan DBSCAN, yaitu perlakuan terdapat data yang mengandung noise. Untuk itu maka penelitian ini mempelajari pengaruhi reduksi dimensi pada data berdimensi tinggi terhadap hasil pengklasteran oleh algoritma k-Means yang mewakili metode centroid dengan hasil pengklasteran oleh algoritma DBSCAN yang mewakili metode density. Sekalipun secara numerikal terjadi peningkatan kualitas hasil klaster pada k-Means setelah direduksi oleh Singular Value Decomposition (SVD), yaitu dari rata-rata jarak awal 1.04136 menjadi 0.003, namun secara statistik perubahan tersebut tidak signifikan atau dianggap sama. Sehingga secara statistik dapat disimpulkan bahwa tidak ada pengaruh oleh SVD terhadap kualitas hasil klaster oleh k-Means. Sebaliknya pada DBSCAN pengaruh reduksi dimensi oleh SVD sangat signifikan dimana mampu mengubah

kualitas hasil klaster dari semula rata-rata jarak intra klaster 76.13480 menjadi 13.71130 atau terjadi peningkatan kualitas 555.27%. Pengaruh yang signifikan oleh SVD juga ditunjukkan pada perubahan waktu komputasi klaster baik pada optimasi SVD + k-Means dan terlebih lagi optimasi SVD + DBSCAN. Optimasi SVD mampu mempercepat waktu komputasi k-Means dari 3.68182 second menjadi 2.09091 second atau 1.76 kali lebih cepat. Sementara itu optimasi SVD mempercepat waktu komputasi DBSCAN dari 19.40000 second menjadi 0.97500 second atau 19.89 kali lebih cepat.

Pendahuluan

Clustering adalah suatu metode dalam data mining, yaitu suatu metode data mining yang mengolah kumpulan data menjadi kumpulan data yang lebih kecil atau mengelompokkan data berdasarkan kesamaan atribut data. Dengan demikian pengklasteran data bersifat unsupervised learning [1, 2]. Proses pengklasteran yang unsupervised tersebut diejawantahkan dengan mengukur jarak antar data. Mekanisme pengukuran jarak tersebut ada yang menggunakan pendekatan metode berbasis centroid [1] dan juga yang menggunakan pendekatan metode berbasis density (kepadatan).

Algoritma yang sering digunakan dalam pengklasteran adalah algoritma k-Means, karena mudah digunakan dan dibandingkan dengan algoritma pengelompokan lainnya, waktu pemrosesan lebih cepat. Algoritma k-Means termasuk jenis pengklasteran partisi karena akan mengkelompokkan data ke dalam sejumlah k kelompok, dimana k adalah jumlah kelompok yang diinginkan [1, 3]. Pada metode ini setiap klaster memiliki titik pusat (centroid) dan secara umum metode ini memiliki fungsi tujuan yaitu meminimumkan jarak (dissimilarity) dari seluruh data ke pusat klaster masing-masing.

Metode pendekatan berbasis kepadatan bekerja dengan mendeteksi area di mana data terkonsentrasi dan dipisahkan oleh area yang kosong atau jarang. Data yang bukan merupakan bagian dari klaster diberi label sebagai noise. Metode ini menggunakan algoritma pengklasteran pembelajaran mesin unsupervised yang secara otomatis mendeteksi pola berdasarkan lokasi spasial dan jarak ke sejumlah tetangga tertentu. Algoritma ini menemukan kelompok fitur data dalam noise sekitar berdasarkan distribusi spasialnya. Menggunakan jarak yang ditentukan untuk memisahkan klaster padat dari noise yang lebih jarang. Algoritma DBSCAN adalah yang tercepat dari metode kepadatan, tetapi hanya sesuai jika terdapat jarak pencarian (search distance) yang sangat jelas untuk digunakan, dan berfungsi dengan baik untuk semua klaster potensial. Ini mengharuskan semua klaster yang bermakna memiliki kepadatan yang sama.

Data berdimensi tinggi merupakan data yang mempunyai jumlah fitur banyak. Setiap kata yang berbeda dalam dokumen teks akan menjadi atribut baru setelah dikonversi menjadi data numerik sehingga data memiliki banyak dimensi [4]. Pada data berdimensi tinggi dapat muncul data yang kompleks, yaitu data dengan noise, anomali (outlier), kehilangan makna (missing values), dan diskontinuitas antar data [5, 6]. Beberapa contoh data berdimensi tinggi adalah data dokumen teks, data gambar, dan data ekspresi gen. Dalam proses clustering dokumen teks, setiap dimensi atau fitur data memiliki pengaruh yang penting. Ketika data dikelompokkan ke dalam kelas, setiap karakteristik data mempengaruhi lokasi data. Hanya saja ketika sebuah data memiliki banyak fitur dan terlalu terdiversifikasi, penelitian sebelumnya menunjukkan bahwa algoritma clustering seperti algoritma k-Means tidak dapat menemukan kedekatan atau kesamaan antar data, sehingga data tidak dapat dikelompokkan dengan benar.

Ini sering disebut sebagai curse of dimensionality. Untuk mendapatkan hasil yang baik, data berdimensi tinggi harus melalui tahap pengolahan awal atau preprocessing yaitu reduksi dimensi. Pengurangan dimensi adalah proses mereduksi variabel acak atau variable yang tidak penting dengan pertimbangan tertentu [1, 6, 7]. Untuk itu, diperlukan penelitian untuk memperoleh metode reduksi dimensi yang lebih baik. Ada dua metode untuk reduksi dimensi, yaitu seleksi fitur dan ekstraksi fitur. Dari penelitian sebelumnya, reduksi dimensi menggunakan metode seleksi fitur berpengaruh baik terhadap hasil clustering menggunakan algoritma k-Means [6- 8]. Pada penelitian ini digunakan algoritma Singular Value Decomposition (SVD). Terdapat perbedaan perilaku yang signifikan antara k-Means dan DBSCAN, yaitu perlakuan terdapat data yang mengandung kebisingan (noise) dan anomali (outlier). Untuk itu maka penelitian ini mempelajari pengaruhi reduksi dimensi SVD pada data berdimensi tinggi terhadap hasil pengklasteran oleh algoritma k-Means yang mewakili metode centroid dengan hasil pengklasteran oleh algoritma DBSCAN yang mewakili metode density.

Reduksi dimensi merupakan teknik dalam text mining dengan mengurangi dimensi sehingga clustering memproses data dengan jumlah fitur yang telah berkurang. SVD adalah metode populer untuk reduksi dimensi. Untuk data tekstual, metode ini juga dikenal dengan Latent Semantic Analysis (LSA) [9, 10]. SVD berpengaruh terhadap clustering k-Means dengan meningkatkan kualitas hasil pengklasteran dan meningkatkan kecepatan waktu komputasi [6-8]. Namun k-Means memiliki kelemahan yaitu k- Means kurang baik jika bekerja pada data besar serta memiliki noise dan outlier [11, 12]. Sebaliknya DBSCAN justru bekerjanya dengan memisahkan data berdasarkan kepadatan, dimana data padat terkonsentasi (jarak antar data dekat) versus data yang jarang atau berjarak jauh yang dikenali sebagai noise atau outlier. Sementara SVD adalah pereduksi jenis feature selection yang memilih atribut terbaik dan membuang atribut yang dianggap kurang bernilai. Untuk itu maka perlu diteliti bagaimana pengaruh SVD terhadap kedua jenis metode pengklasteran tersebut.

Rumusan masalah pada penelitian ini adalah apa pengaruh reduksi dimensi SVD terhadap hasil pengklasteran dokumen teks berbahasa Indonesia menggunakan pendekatan metode centroid dan metode density. SVD adalah state of the art daripada teknik reduksi dimensi yang sudah terbukti meningkatkan kualitas hasil pengklasteran dan waktu komputasi pada k-Means sarat dengan persyaratan karena rentan terhadap noise dan outlier. Pengujian dan perbandingan terhadap kualitas hasil pengklasteraan oleh algoritma DBSCAN dengan pendekatan metode kepadatan, secara khusus perlu dilakukan pada data uji yang berasal dari dokumen teks berbahasa Indonesia. Hal ini tujuan dapat memperkaya pemahaman teks mining yang sesuai dengan kaedah tata bahasa Indonesia.

Landasan Teori

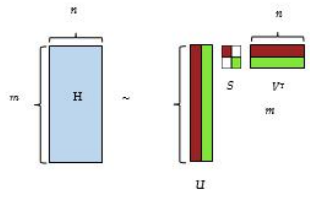
Pada data berdimensi tinggi, biasanya hanya beberapa dimensi yang diasosiasikan dengan cluster tertentu, tetapi data dengan dimensi non-esensial dapat membuat banyak noise dan cluster yang tidak jelas yang seharusnya terbentuk. Selain itu, dengan bertambahnya ukuran, data cenderung semakin menyusut karena titik data biasanya berada dalam dimensi subruang yang berbeda. Jika data Anda sangat berjauhan, Anda dapat mengasumsikan bahwa titik data dalam dimensi yang berbeda berjarak sama. Menghitung jarak yang penting untuk clustering tidak ada artinya [6]. Reduksi dimensi merupakan rangkaian proses penting yang harus dilakukan Ketika melakukan clustering data berdimensi tinggi [13]. Data berdimensi tinggi adalah data dengan banyak atribut atau karakteristik yang berbeda.

Gejala ini menyebabkan kesalahan saat mengelompokkan data. Hal ini dikarenakan sulitnya mencari kesamaan pada setiap datadan kualitas hasil cluster yang kurang baik. Pendekatan umum untuk memecahkan masalah dengan data berdimensi tinggi. Salah satunya adalah pengurangan dimensi [14].

Alasan utama pengurangan dimensi adalah untuk memasukkan data dimensi tinggi ke dalam data dimensi rendah. Fitur berdimensi rendah dianggap sebagai fitur yang paling penting atau paling penting.

Pengurangan dimensi dapat dibagi menjadi dua metode: seleksi fitur dan ekstraksi fitur. Metode reduksi dimensi yang sudah dikenal

secara luas salah satunya adalah Latent Semantic Analysis atau lebih dikenal sebagai Singular Value Decomposition. Reduksi dapat dicapai dengan memetakan kumpulan data dari dimensi asli ke dimensi lain yang relatif lebih rendah untuk menangkap properti data. Pemetaan ini mengekstrak komponen atau fitur dari dimensi baru yang memiliki dampak signifikan pada kumpulan data dan membuat prinsip komponen untuk membuang data yang tidak berdampak[1, 6]. Proses reduksi dimensi diterapkan sebelumproses clustering dan setelah preprocessing danword weighting SVD adalah suatu bentuk analisa faktor pada matriks. Dalam SVD, matriks dipecah menjadi tiga komponen matriks berdasarkan frekuensi kemunculan kata kunci [15]. Komponen matriks pertama (U) menggambarkan entitas baris sebagai vektor ortogonal dari matriks. Komponen matriks kedua (S) adalah matriks diagonal yang memuat nilai-nilai scalar matriks. Dan komponen ketiga (V) adalah matriks entitas kolom sebagai matriks vector ortogonal. Dalam penelitian ini, SVD dari matriks tf-idf digunakan dalam pendekatan transformasi linier. Tujuan utama dari SVD adalah untuk memperkirakan peringkat matriks. Gambar 1 menunjukkan SVD.



Gambar 1. Komponen Singular Value Decomposition

Jika dimensi matriks H adalah m×n dan nilai m ≥ n serta rank (H) = r maka melalui persamaan singular value decomposition dari H, didefinisikan:

$$H = U\,S\,V^T \quad (1)$$

Keterangan:

H	: matriks TF-IDF
U	: vektor singular kiri
V	: vektor singular kanan
T	: transpose
S	: nilai singular
m	: dokumen
n	: term
r	: rank

$$\text{dengan, } U^T U = V^T V = I_n \quad (2)$$

$$\text{serta terpenuhi kondisi, } S = \text{diag}(\sigma_i, \cdots, \sigma_n) \quad (3)$$

$$\text{untuk, } \sigma_i > 0 \text{ untuk } 1 \le i \le r \quad (4)$$

$$\sigma_j = 0 \text{ untuk } j \ge r + 1 \quad (5)$$

Kolom pertama dari matriks U dan V mendefinisikan vektor eigen ortonormal yang sesuai dengan nilai r bukan nol vektor eigen dari masing-masing matriks HHT dan HTH. Kolom-kolom matriks U dan V masing-masing memuat vektor-vektor yang disebut vector singular kiri dan kanan. Nilai singular dari H adalah elemen diagonal dari matriks S, dan nilai singular diambil dari akar kuadrat dari nilai atribut dari beberapa nilai eigen dari HHT. Setelah mendapatkan tiga matriks dari proses SVD, maka proses selanjutnya untuk mereduksi dimensi matriks adalah dengan mereduksi dimensi matriks S berupa matriks diagonal. Kolom-kolom matriks U dan barisbaris matriks V juga menghilang setelah posisi nilai singular pada matriks S karena nilai scalar matriks terkecil milik matriks S dihilangkan. Kemudian kalikan matriks baru U dan S untuk membuat matriks baru H, yaitu matriks baruberisi data komponen utama.

K-Means adalah metode pemisahan yang membagi data menjadi k cluster. K-Means meminimalkan jarak total data individu dalam cluster melalui proses partisi berulang [16]. Hasil pengklasteran menggunakan algoritma k-Means sangat tergantung pada pilihan centroid atau center point. Titik tengah ini menentukan hasil pengklasteran [17]. Selanjutnya, jarak Euclidean digunakan untukmenghitung jarak antara data dan pusat gravitasi. Elemen data terjauh dari centroid terkandung dalam cluster lain yang paling dekat dengan elemen itu. Langkah-langkah untuk algoritma kmeans [18] adalah sebagai berikut:

a) Menetapkan nilai k sebagai jumlah klister yang ingin dihasilkan.

b) Temukan nilai centroid awal dari kMeans secara acak. Gunakan rumus jarak Euclidean untuk menghitung setiap jarak dari setiap data ke setiap pusat gravitasi. Jika Anda menulisnya dalam rumus matematika, itu akan menjadi sebagai berikut:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2} \quad (6)$$

Keterangan :
d_{xy} = tingkat perbedaan (dissimilarity degree)
n = jumlah kata
x_i = centroid cluster ke-i
y_i = data vector

Tentukan lokasi centroid baru dengan rata-rata data pada centroid yang sama. Hitung jarak dari setiap titik data ke centroid baru menggunakan rumus jarak Euclidean. Jika ada perubahan data centroid asli dan centroid baru, kembali ke langkah 4. Algoritma berhenti ketika tidak ada perubahan pada data co-center (berada pada centroid yang sama). DBSCAN adalah algoritma pengelompokan berbasis kepadatan data. Setiap objek di area radius harus berisi data minimal. Objek apa pun yang tidak termasuk dianggap kebisingan. Perhitungan algoritma DBSCAN adalah:

a) Tetapkan parameter min-pts dan eps.

b) Tetapkan random titik awal atau p.

c) Lakukan perulangan untuk semua titik diproses sampai selesai.

d) Kalkulasi nilai eps untuk semua jarak ke titik yang dapat dijangkau dengan kepadatan hingga p menggunakan persamaan:

$$E(x,y) = \sqrt{\sum_{i=0}^{n}(X_i - Y_i)^2} \quad (7)$$

e) Jika titik yang memenuhi eps lebih dari min-pts maka titik p adalah core-point dan klaster terbentuk.

f) Jika p adalah border-point dan tidak ada titik yang density reachable terhadap p, maka proses dilanjutkan ke titik lain

Perbandingan secara langsung terhadap hasil reduksi dimensi data tidak dapat dilakukan, untuk itu diperlukan perbandingan lain sebagai representasi dari kualitas hasil reduksi. Merujuk kepada teori dan konsep curse of dimensionality bahwa semakin tinggi dimensi data maka berdampak semakin rendahnya kualitas hasil pengklasteran. Sehingga semakin baik hasil pereduksi dimensi data maka menghasilkan klaster data yang berkualitas lebih baik [1, 2, 6, 8].

Validasi hasil pengklasteran yang paling ideal adalah validasi internal karena sesuai dengan tujuan pengklasteran, yaitu untuk mengelompokkan objek ke dalam cluster yang identik atau paling mirip dan mengelompokkan objek yang berbeda ke dalam cluster yang berbeda.

Langkah-langkah untuk memeriksa pengelompokan internal didasarkan pada dua kriteria. Yang pertama adalah compactness yaitu elemen-elemen dari setiap klaster harus saling berdekatan. Ukuran unik dari kekompakan adalah disperse variannya. Yang kedua separateness yaitu dirancang untuk mengukur seberapa berbeda atau terisolasinya suatu klaster dari klister lainnya [19]. Salah satu perhitungan untuk verifikasi hasil klastering internal yang digunakan untuk mengukur penempatan objek dalam klister adalah Silhouette Coefficient. Tahapan perhitungan Silhouette Coefficient adalah sebagai berikut:

a) Dengan Euclidean Distance dapatkan nilai rata-rata jarak dari suatu data dengan semua data lain yang berada dalam suatu klaster.

b) Dengan cara yang sama dapatkan nilai rata-rata jarak dari suatu data dengan semua data di klaster lain, selanjutnya diambil nilai terkecilnya.

c) Nilai Silhouette Coefficient adalah:

$$s(i) = \frac{b(i)-a(i)}{\text{ma}\ (a(i),b(i))} \quad (8)$$

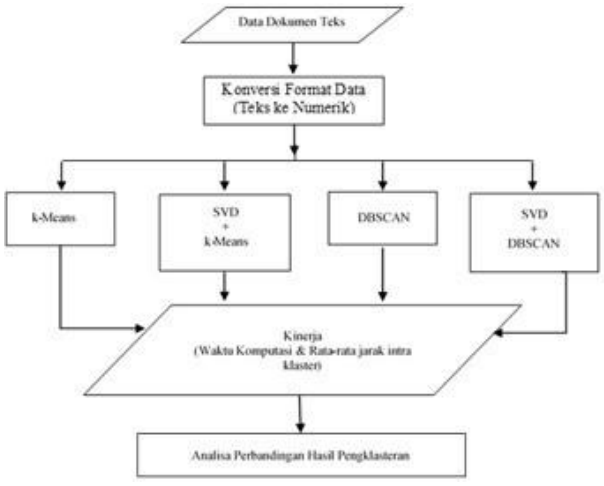
Keterangan:
s(i) : Silhouette Coefficient.

a(i) : nilai rata-rata jarak dari data dengan data lainnya yang berada dalam klaster yang sama.
b(i) : nilai rata-rata jarak dari data dengan data lainnya yang berada dalam klaster yang berbeda

Metode

Untuk melakukan perbandingan pada hasil reduksi dimensi algoritma SVD pada hasil pengklasteran algoritma k-Means dan algoritma DBSCAN, maka tahapan-tahapan penelitian adalah sebagaimana ditunjukkan pada Gambar 2.

Data uji akan melalui tahapan konversi format data, penentuan jumlah klaster yang ideal, pengklasteran dilakukan pada data yang masih berdimensi tinggi menggunakan k-Means dan DBSCAN. Kemudian data direduksi dimensinya menggunakan SVD. Data yang sudah telah dikurangi jumlah atributnya selanjutnya diklaster dengan menggunakan algoritma k-Means dan DBSCAN. Pada setiap proses dilakukan pengamatan terhadap luaran proses dengan mencatat jumlah waktu komputasi dan nilai rata-rata jarak intra klister pada setiap proses.



Gambar 2. Tahapan Penelitian

Pada proses konversi data dokumen teks, langkah pertama yang dilakukan adalah case folding, proses ini menyeragamkan semua huruf alfabet kapital menjadi huruf kecil. Selanjutnya dilakukan tokenizing yang merupakan proses memecah kalimat menjadi potongan kata. Langkah ketiga, potongan kata-kata hasil tokenizing dilakukan stop words removal yang berfungsi menghilangkan kata-kata memiliki makna. Berikutnya adalah stemming yaitu mengubah kata yang memiliki imbuhan kembali ke bentuk dasarnya menggunakan metode Nazief Adriani [20].

Langkah terakhir dari proses konversi data teks menjadi numerik adalah pembobotan kata dengan menggunakan metode gabungan term frequency (TF) dan inverse document frequency (IDF) yang dikenal dengan istilah TF-IDF. Reduksi dimensi yang dilakukan dengan algoritma SVD dilakukan pada matriks hasil pembobotan kata. Dengan demikian maka, ukuran matriks hasil pembobotan kata akan lebih kecil sehingga mempermudah proses pengklasteran serta memberikan hasil kualitas klasteran yang lebih baik. Validasi internal hasil klaster digunakan untuk melihat kualitas dan kekuatan klaster. Yaitu dengan membandingkan nilai rata-rata jarak dari suatu data dengan semua data lain yang berada dalam klaster yang sama dengan nilai rata-rata jarak klaster yang berbeda.

Perhitungan menggunakan Euclidean distance, dan diambil nilai terkecilnya. Semakin baik hasil pengklasteran ditunjukkan oleh semakin kecil rata-rata jaraknya. Dengan demikian,

meminimalkan indeks, maka kelompok klister akan berbeda dari klaster lainnya sehingga mencapai proses partisi yang baik. Pada penelitian ini indikator capaian berupa kriteria pengujian. Pada pengujian awal diperlukan indikator capaian awal yaitu mampu menetapkan jumlah klaster optimal untuk data asal (original) yang masih berdimensi tinggi, dan parameter untuk algoritma DBSCAN. Untuk menentukan nilai k optimal untuk algoritma k-Means, mengacu pada nilai ideal hasil evaluasi internal klister dengan rata-rata jarak, yang selanjut jumlah klaster optimal tersebut juga akan digunakan untuk pengklasteran data hasil reduksi dimensi.

Hasil dan Pembahasan

Jumlah klaster khususnya pada algoritma kMeans ditentukan berdasarkan jumlah yang diinginkan, atau dapat pula dengan mencari jumlah klaster yang paling ideal sesuai dengan kondisi data yang digunakan [21]. Jumlah klaster yang optimum akan terpenuhi apabila kondisi data yang mengerombol bersifat homogen terhadap data lain yang ada pada satu klaster yang sama, dan bersifat heterogeny terhadap data dalam kelompok berbeda. Berdasarkan hal tersebut, pengujian ini menggunakan nilai intra klaster atau jarak rata-rata setiap titik pusat klaster terhadap titik pusat klaster lain sebagai metode evaluasi. Dalam menentukan nilai k yang optimal, digunakan nilai rata-rata jarak data dari centroid (intra-cluster). Tes dilakukan dengan menggunakan nilai k yang bervariasi untuk melihat perubahan dalam hasil pengelompokan. Dari beberapa nilai k yang didapatkan, penentuan nilai k yang optimal menggunakan metode elbow. Metode ini menentukan jumlah kluster dengan melihat perubahan nilai yang paling signifikan dibandingkan dengan jumlah cluster lain [22]. Yaitu pada nilai k dimana terjadi perubahan nilai paling besar dalam perbandingan dengan jumlah klaster yang lain. Sehingga dengan demikian dapat disimpulkan bahwa pada nilai k tersebut adalah jumlah klaster paling ideal dari data. Semua proses pengklasteran baik yang tanpa reduksi maupun dengan metode reduksi dimensi, dilakukan dengan algoritma k-Means dengan menggunakan k=4. Dalam pengujian pengklasteran dengan DBSCAN, dilakukan pada berbagai kondisi parameter MinPoint (dari 2 sd 5) dan Epsilon (dari 0.10 sd 1.00). Uji pengklasteran dilakukan pada data berdimensi tinggi dengan dilakukan secara langsung dengan alogoritma k-Means pada nilai k=4. Sementara untuk data berdimensi rendah dilakukan reduksi dimensi dengan metode SVD terlebih dahulu. Data hasil pengujian berupa nilai evaluasi kualitas klister dengan rata-rata jarak intra klaster, waktu komputasi pengklasteran, serta tambahan waktu reduksi. Data hasil pengujian selanjutnya dianalisa menggunakan perangkat lunak uji statistic yaitu SPSS versi 22.0. Untuk memastikan bahwa data dapat diuji dengan Teknik parametrik maka seluruh data terlebih dahulu diuji Normalitas dan Homagenitas, jika tidak memenuhi kaedah normalitas maka data tersebut selanjutnya diuji dengan Teknik non-parametrik.

Tabel 1 menunjukkan nilai rata-rata beserta nilai standar deviasi dan nilai varian daripada hasil pengklasteran pada data dimensi tinggi oleh algoritma k-Means dan DBSCAN. Secara statistik deskriptif terlihat bahwa algoritma kMeans mampu memberikan hasil yang lebih baik dari pada DBSCAN baik pada kualitas hasil pengklasteran yang ditunjukkan dengan lebih rendahnya rata-rata jarak intra klister, yang diikuti dengan lebih rendahnya nilai standar deviasi dan varian yang menunjukkan stabilitas hasil pengklasteran sekali pun dilakukan berulang-kali.

Tabel 1. Hasil Pengklasteran k-Means pada data dimensi tinggi

Nama Algoritma Klastering	Nilai Rata-rata Intra Klaster			Waktu Komputasi		
	Mean	Standar Deviasi	Varian	Mean	Standar Deviasi	Varian
k-Means	1.041	0.13	0.00	3.68	1.13	1.27
DBSCAN	76.135	15.208	231.283	19.40	2.82	7.94

Pada Tabel 2, dikarenakan semua kelompok data tidak memenuhi kaedah normalitas dan homogenitas maka uji beda dilakukan dengan metode non-parametrik. Uji perbedaan menggunakan Uji Mann-Whitney, terbukti bahwa terdapat beda yang sangat signifikan baik pada nilai rata-rata jarak intra klister maupun waktu komputasi, bahwa algoritma k-Means lebih baik daripada DBSCAN.

Tabel 2. Hasil Uji Beda Mann-Whitney Hasil Klaster Data Dimensi Tinggi

	Nilai Rata-rata Intra Klaster	Waktu Komputasi
Mann-Whitney U	.000	.000
Wolcoxon W	253.00	253.00
Z	-6.764	-6.524
Asymp. Sig. (2-tailed)	.000	.000

Secara statistik deskriptif terlihat bahwa algoritma k-Means mampu memberikan hasil yang lebih baik dari pada DBSCAN baik pada kualitas hasil pengklasteran yang ditunjukkan dengan lebih rendahnya rata-rata jarak intra

klister, yang diikuti dengan lebih rendahnya nilai standar deviasi dan varian yang menunjukkan stabilitas hasil pengklasteran sekali pun dilakukan berulang-kali. Namun waktu komputasi DBSCAN menjadi lebih cepat daripada waktu komputasi k-Means. Hal ini menunjukkan bahwa DBSCAN sangat rentan terhadap data berdimensi tinggi, sehingga komputasinya menjadi lebih baik pada data berdimensi rendah, sekalipun kualitas hasil belum sebaik k-Means.

Tabel 3. Hasil Pengklasteran Data Dimensi Rendah

Nama Algoritma Reduksi-Klastering	Nilai Rata-rata Intra Klaster			Waktu Komputasi		
	Mean	Standar Deviasi	Varian	Mean	Standar Deviasi	Varian
SVD + k-Means	.003	.000	.000	2.09	1.15	1.32
SVD + DBSCAN	13.711	1.670	2.789	.98	.92	.85

Tabel 3 menunjukkan nilai rata-rata beserta nilai standar deviasi dan nilai varian daripada hasil pengklasteran pada data dimensi rendah oleh algoritma k-Means dan DBSCAN, dimana data yang semula berdimensi tinggi telah direduksi terlebih dahulu oleh algoritma SVD. Pada Tabel 4, uji perbedaan digunakan Uji Mann-Whitney, terbukti bahwa terdapat beda yang sangat signifikan baik pada nilai rata-rata jarak intra klister maupun waktu komputasi, bahwa algoritma k-Means lebih baik daripada DBSCAN.

Tabel 4. Hasil Uji Beda Mann-Whitney Hasil Klaster Data Dimensi Rendah

	Nilai Rata-rata Intra Klaster	Waktu Komputasi
Mann-Whitney U	.000	194.000
Wolcoxon W	253.000	1014.000
Z	-7.428	-3.816
Asymp. Sig. (2-tailed)	.000	.000

Untuk dapat melihat pengaruh dari reduksi dimensi terhadap hasil pengklasteran oleh k-Means dan DBSCAN, maka dilakukan uji perbandingan terhadap 4 kelompok data yaitu oleh k-Means, DBSCAN, SVD+k-Means, dan SVD+DBSCAN. Tabel 5 menjelaskan secara deskriptif kondisi data hasil pengklasteran untuk setiap kelompok baik pada rata-rata jarak intra klaster maupun waktu komputasi klaster. Uji perbandingan Analysis of Variance (ANOVA) pada tabel 6 menunjukkan pada dari 4 kelompok data tersebut ada yang berbeda secara signifikan baik pada rata-rata jarak intra klaster mau pun pada waktu komputasi klaster. Uji lanjutan Post Hoc dengan uji Tukey HSD pada Tabel 7 serta Homogeneous Subsets pada Tabel 8 menjelaskan secara rinci perbedaan tersebut. SVD mampu meningkatkan kinerja kMeans dan DBSCAN. Kenyataan ini mendukung teori tentang curse of dimensiolity. Hal yang menarik terlihat bahwa sekalipun secara numerikal terjadi peningkatan kualitas hasil klaster pada k-Means setelah direduksi oleh SVD, yaitu dari rata-rata jarak awal 1.04136 menjadi 0.003, namun secara statistic perubahan tersebut tidak signifikan atau dianggap sama. Sehingga secara statistik dapat disimpulkan bahwa tidak ada pengaruh oleh SVD terhadap kualitas hasil klaster oleh k-Means. Sebaliknya pada DBSCAN pengaruh reduksi dimensi oleh SVD sangat signifikan dimana mampu mengubah kualitas hasil klaster dari semula rata-rata jarak intra klister 76.13480 menjadi 13.71130 atau terjadi peningkatan kualitas 555.27%. Pengaruh yang signifikan oleh SVD juga ditunjukkan pada perubahan waktu komputasi klaster baik pada optimasi SVD + k-Means dan terlebih lagi optimasi SVD + DBSCAN. Optimasi SVD mampu mempercepat waktu komputasi k-Means dari 3.68182 second menjadi 2.09091 second atau 1.76 kali lebih cepat. Sementara itu optimasi SVD mempercepat waktu komputasi DBSCAN dari 19.40000 second menjadi 0.97500 second atau 19.89 kali lebih cepat.

Tabel 5. Statistik Deskriptif Perbandingan Hasil Klaster

Objek Pengujian	Algoritma	N	Mean	Standard Deviation	Standard Error	95% Confidence Interval for Mean		Minimum	Maximum
						Lower Bound	Upper Bound		
Nilai Rata-rata Intra Klaster	k-Means	22	1.04136	.013272	.002830	1.03548	1.04725	1.013	1.069
	SVD + k-Means	22	.00300	.000000	.000000	.00300	.00300	.003	.003
	DBSCAN	40	76.13480	15.207991	2.404594	71.27105	80.99855	42.480	85.891
	SVD + DBSCAN	40	13.71130	1.670108	.264067	13.17717	1424543	8.764	14.261
Waktu Komputasi	k-Means	22	3.68182	1.129111	.240727	3.18120	4.18244	2.000	6.000
	SVD + k-Means	22	2.09091	1.150945	.245382	1.58061	2.60121	1.000	5.000
	DBSCAN	40	19.40000	2.817528	.445490	18.49891	20.30109	14.000	31.000
	SVD + DBSCAN	40	.97500	8.423659	.145389	.68092	1.26908	.000	3.000

Tabel 6. Uji One Way ANOVA

		Sum of Squares	df	Mean Square	F	Sig.
Nilai Rata-rata Intra Klaster	Between Groups	133909.103	3	44636.368	586.753	.000
	Within Groups	9128.821	120	76.074		
	Total	143037.924	123			
Waktu Komputasi	Between Groups	8330.673	3	2776.891	839.012	
	Within Groups	397.166	120	3.310		
	Total	8727.839	123			

Tabel 7. Uji Post Hoc Tukey HSD Multiple Comparisons

Dependent Variable	I (Nama Algoritma)	J (Nama Algoritma)	Mean Difference (I-J)	Standard Error	Sig.	95% Confidence Interval for Mean	
						Lower Bound	Lower Bound
Average Intra Cluster	k-Means	SVD + k-Means	1.038364	2.629786	.979	-5.81327	7.89000
		DBSCAN	-75.093436*	2.315108	.000	-81.12521	-69.06166
		SVD + DBSCAN	-12.669936*	2.315108	.000	-18.70171	-6.63816
	SVD + k-Means	k-Means	-1.038364	2.629786	.979	-7.89000	5.81327
		DBSCAN	-76.131800*	2.315108	.000	-82.16358	-70.10002
		SVD + DBSCAN	-13.708300*	2.315108	.000	-19.74008	-7.67652
	DBSCAN	k-Means	75.093436*	2.315108	.000	69.06166	81.12521
		SVD + k-Means	76.131800*	2.315108	.000	70.10002	82.16358
		SVD + DBSCAN	62.423500*	1.950301	.000	57.34219	67.50481
	SVD + DBSCAN	k-Means	12.669936*	2.315108	.000	6.63816	18.70171
		SVD + k-Means	13.708300*	2.315108	.000	7.67652	19.74008
		DBSCAN	-62.423500*	1.950301	.000	-67.50481	-57.34219
Clustering Computation Time	k-Means	SVD + k-Means	1.590909*	.548528	.023	.16177	3.02004
		DBSCAN	-15.718182*	.482892	.000	-16.97631	-14.46006
		SVD + DBSCAN	2.706818*	.482892	.000	1.44869	3.96494
	SVD + k-Means	k-Means	-1.590909*	.548528	.023	-3.02004	-.16177
		DBSCAN	-17.309091*	.482892	.000	-18.56722	-16.05097
		SVD + DBSCAN	1.115909	.482892	.101	-.14222	2.37403
	DBSCAN	k-Means	15.718182*	.482892	.000	14.46006	16.97631
		SVD + k-Means	17.309091*	.482892	.000	16.05097	18.56722
		SVD + DBSCAN	18.425000*	.406799	.000	17.36513	19.48487
	SVD + DBSCAN	k-Means	-2.706818*	.482892	.000	-3.96494	-1.44869
		SVD + k-Means	-1.115909	.482892	.101	-2.37403	.14222
		DBSCAN	-18.425000*	.406799	.000	-19.48487	-17.36513

Tabel 8. Homogeneous Subsets Uji Post Hoc Tukey HSD

Nilai Rata-rata Intra Klaster					Waktu Komputasi				
Nama Algoritma	N	Subset for alpha = 0.05			Nama Algoritma	N	Subset for alpha = 0.05		
		1	2	3			1	2	3
SVD + k-Means	22	.00300			SVD + DBSCAN	40	.97500		
k-Means	22	1.04136			SVD + k-Means	22	2.09091		
SVD + DBSCAN	40		13.71130		k-Means	22		3.68182	
DBSCAN	40			76.13480	DBSCAN	40			19.40000
Sig.		.970	1.000	1.000	Sig.		.101	1.00	1.000

Kesimpulan dan Saran

Penelitian ini telah membandingkan hasil pengklasteran dokumen teks dengan algoritma k-Means sebagai algoritma berbasis centroid dan algoritma DBSCAN sebagai algoritma berbasis densitas, serta pengaruh daripada reduksi dimensi oleh SVD. Perbandingan dilakukan pada kualitas hasil klastering yaitu dengan mengukur rata-rata jarak intra klister dan waktu komputasi dari setiap algoritma pengklasteran. Secara umum dalam penelitian ini dapat membuktikan teori tentang curse of dimensionality dimana data berdimensi tinggi harus direduksi terlebih dahulu jika ingin mendapatkan hasil pengkalsteran yang lebih baik, berlaku untuk kualitas hasil klaster, dan waktu komputasi klaster. Kesimpulan ini diperoleh dari hasil uji perbandingan pengklasteran data dimensi tinggi dengan data dimensi rendah, baik oleh k-Means mau pun oleh DBSCAN. Sekali pun secara numerikal terlihat bahwa kualitas hasil klaster k-Means lebih baik daripada kualitas hasil klaster DBSCAN, namun secara statistik terbukti bahwa reduksi dimensi oleh SVD lebih besar pengaruhnya terhadap DBSCAN.

Referensi

[1] [1]J. Han, J. Pei, and M. Kamber, Data mining: concepts and techniques. Elsevier, 2011.

[2] E. Alpaydin, Introduction to machine learning. MIT press, 2014.

[3] X. Jin and J. Han, "K-medoids clustering," Encyclopedia of Machine Learning and Data Mining, pp. 697-700, 2017.

[4] S. Jun, S.-S. Park, and D.-S. Jang, "Document clustering method using dimension reduction and support vector clustering to overcome sparseness," Expert Systems with Applications, vol. 41, no. 7, pp. 3204-3212, 2014.

[5] T. C. Chen, S. Sanga, T. Y. Chou, V. Cristini, and M. E. Edgerton, "Neural network with k-means clustering via pca for gene expression profile analysis," in 2009 World Congress on Computer Science and Information Engineering, 2009: IEEE, pp. 670-673.

[6] M. I. Jambak, F. Mohammed, N. Hidayati, R. Efendi, and R. Primartha, "The Impacts of Singular Value Decomposition Algorithm Toward Indonesian Language Text Documents Clustering," in International Conference of Reliable Information and Communication Technology, 2018: Springer, pp. 173-183.

[7] M. I. Jambak and A. I. I. Jambak, "Comparison of dimensional reduction using the Singular Value Decomposition Algorithm and the Self Organizing Map Algorithm in clustering result of text documents," in IOP Conference Series: Materials Science and Engineering, 2019,vol. 551, no. 1: IOP Publishing, p. 012046.

[8] S. I. R. Hasanah, M. I. Jambak, and D. M. Saputra, "Comparison of Dimensional Reduction Using Singular Value

Decomposition and Principal Component Analysis for Clustering Results of Indonesian Language Text Documents," in The 2nd International Conference of Applied Sciences, Mathematics, & Informatics (ICASMI) 2018, Bandar Lampung, Indonesia, 2018: Universitas Lampung.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," Journal of the American society for information science, vol. 41, no. 6, p. 391, 1990.

[10] S. T. Dumais, "Latent semantic analysis," Annual Review of Information Science and Technology, vol. 38, no. 1, pp. 188-230, 2004, doi: 10.1002/aris.1440380105.

[11] L. Kaufman and P. Rousseeuw, "Clustering by means of medoids. in 'Y. Dodge (editor) Statistical Data Analysis based on L1 Norm', 405-416," ed: Elsevier/North-Holland, 1987.

[12] T. S. Madhulatha, "Comparison between k-means and k-medoids clustering algorithms," in Advances in Computing and Information Technology: Springer, 2011, pp. 472-481.

[13] I. Assent, "Clustering high dimensional data," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 4, pp. 340-350, 2012.

[14] X.-S. Yang, S. Lee, S. Lee, and N. Theera-Umpon, "Information analysis of high-dimensional data and applications," Mathematical Problems in Engineering, vol. 2015, 2015.

[15] J. Leskovec, A. Rajaraman, and J. D. Ullman, Mining of massive datasets. Cambridge university press, 2014.

[16] A. Kaushik and S. Ghosh, "A Survey on Optimization Approaches to K-Means Clustering using Simulated Annealing," International Journal of Scientific Engineering and Technology, vol. 3, no. 7, pp. 845-847, 2014.

[17] U. R. Raval and C. Jani, "Implementing and Improvisation of K-means Clustering," Int. J. Comput. Sci. Mob. Comput, vol. 5, no. 5, pp. 72-76, 2016.

[18] R. Dash and R. Dash, "Comparative analysis of K-means and genetic algorithm based data clustering," International Journal of Advanced Computer and Mathematical Sciences, vol. 3, no. 2, pp. 257-265, 2012.

[19] B. Ristevski, S. Loshkovska, S. Dzeroski, and I. Slavkov, "A Comparison of Validation Indices for Evaluation of Clustering Results of DNA Microarray Data," The 2nd International Conference on Bioinformatics and Biomedical Engineering (ICBBE), pp. 587-591, 16-18 May 2008 2008. IEEE.

[20] M. Adriani, J. Asian, B. Nazief, S. M. Tahaghoghi, and H. E. Williams, "Stemming Indonesian: A confix- stripping approach," ACM Transactions on Asian Language Information Processing (TALIP), vol. 6, no. 4, pp. 1-33, 2007.

[21] B. Y. Setia Pramana, Siti Mariyah, Ibnu Santoso, Rani Nooraeni, "DATA MINING dengan R Konsep Serta Implementasi," vol. 1, p. 300, 2018.

[22] M. Syakur, B. Khotimah, E. Rochman, and B. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," in IOP Conference Series: Materials Science and Engineering, 2018, vol. 336, no. 1: IOP Publishing, p. 012017.