Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202072538
Vol. 7, No. 6, Desember 2020, hlm. 1213-1220 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

PENERAPAN ALGORITMA PILLAR UNTUK INISIALISASI TITIK PUSAT K-MEANS KLASTER DINAMIS

Ketut Agus Seputra*[1], I Nyoman Saputra Wahyu Wijaya[2]

[1]Manajemen Informatika, Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha
[2]Ilmu Komputer, Fakultas Teknik dan Kejuruan Universitas Pendidikan Ganesha
Email: [1]agus.seputra@undiksha.ac.id, [2]wahyu.wijaya@undiksha.ac.id
*Penulis Korespondensi

(Naskah masuk: 04 Oktober 2019, diterima untuk diterbitkan: 26 November 2020)

Abstrak

K-Means merupakan algoritma yang digunakan untuk melakukan pengklasteran data. Namun, *k-means* memiliki masalah dalam sensitivitas penentuan partisi awal jumlah klaster. Penelitian terkait menyatakan algoritma *k-means* tergantung pada penentuan titik pusat klaster awal. Pemilihan pusat klaster awal secara acak cenderung menghasilkan klaster yang berbeda. Sehingga untuk menentukan klaster terbaik harus dilakukan dengan memperhatikan nilai *Sum Sequare Error* yang terkecil. Untuk mengatasi permasalahan tersebut, penentuan klaster dilakukan dengan menggunakan algoritma pillar. Algoritma pillar menentukan titik pusat klaster dengan memilih data dengan nilai euclidean paling jauh dari titik pusat klaster. Namun pemilihan titik klaster tetap memperhatikan kemungkinan data outlier. Pengujian dilakukan dengan menetapkan satu buah klaster awal sebagai inisialisasi skaligus sebagai klaster pembanding untuk menentukan kualitas klaster berikutnya. Penelitian ini menggunakan data set ruspini dan iris. Untuk data ruspini terdiri dari 76 data set, sedangkan data iris terdiri dari 150 data set. Klaster *Pillar* memiliki nilai *Sum Sequere Error*, *Variance Cluster*, dan *Davies* yang lebih kecil dibandingkan klaster dinamis pada data set ruspini. Nilai tersebut secara berurutan untuk algoritma pillar adalah 0.28, 0.11, 7.30, 5.88. Untuk data set iris nilai *Sum Square Error* lebih tinggi dibandingkan dengan klaster dinamis yaitu 0.34. Sedangkan algoritma klaster dinamis memiliki nilai 0.32. Hal tersebut disebahkan penentuan data *outlier* pada iris data set yang tidak akurat. Ketidakakurantan tersebut berasal dari data yang bersifat multivariat, sehingga memungkinkan data *outlier* menjadi *centroid* awal klaster. Sehingga jika dilihat dari nilai validitas SSE, algoritma *pillar k-means* klaster dinamis masih kurang bekerja optimal dibandingkan dengan algoritma *k-means* klaster dinamis.

Kata kunci: *klaster, k-means, klaster dinamis, pillar, mean based*

APPLICATION OF PILLAR ALGORITHM FOR INITIALIZATION OF K-MEANS DYNAMIC CLUSTER CENTROID

Abstract

K-Means is an algorithm used to cluster data. However, k-means has a problem in the sensitivity of determining initial partition number of clusters. Related research states the k-means algorithm depends on determining the initial cluster centroid. Random selection of initial cluster centers tends to produce different clusters. So to determine the best cluster must be done by paying attention to the smallest Sum Sequare Error value. To overcome these problems, cluster determination is done using the pillar algorithm. The pillar algorithm determines the cluster centroid by selecting data with the euclidean value farthest from the cluster centroid. However, the selection of centroid still considers the possibility of outlier data. The test is done by assigning one initial cluster as a scalial initialization as a comparison cluster to determine the quality of the next cluster. This study uses Ruspini and iris data sets. The Ruspini data consists of 76 data sets, while the iris data consists of 150 data sets. Pillar cluster has smaller Sum Sequere Error, Variance Cluster, and Davies values than dynamic cluster in the ruspini data set. These values respectively for the pillar algorithm are 0.28, 0.11, 7.30, 5.88. For the iris data set the value of Sum Square Error is higher compared to dynamic cluster which is 0.34. Whereas the dynamic cluster algorithm has a value of 0.32. This is caused by the determination of outlier data in the inaccurate data set slices. This uncertainty is derived from multivariate data, allowing outlier data to be the initial centroid of the cluster. So when viewed from the validity value of the SSE, the dynamic cluster cluster kmeans algorithm still does not work optimally compared to the dynamic cluster k-menas algorithm.

Keywords: *cluster, k-means, dynamic cluster, pillar, mean based*

1. PENDAHULUAN

Clustering atau klaterisasi adalah suatu metode pengelompokan yang mengenakan aturan berdasarkan sekelompok titik atau objek pada data. Dalam berbagai metode klasterisasi, *K-Means* adalah yang paling luas dan sering digunakan (Wu, 2012). Metode ini mengelompokan data berdasarkan jarak antara objek atau titik. K-Means merupakan algoritma pengelompokan yang dikembangkan oleh Mac Queen di tahun 1967 (Barakbah & Kiyoki, 2009). Metode ini cukup sederhana, data set dibagi atau dipartisi kedalam beberapa klaster k. Algoritma ini mudah dijalankan, relatif cepat, mudah disesuaikan sesuai kebutuhan (Aggarwal, Aggarwal, & Gupta, 2012). Namun, k-means memiliki masalah dalam sensitivitas penentuan partisi awal jumlah klister. Dapat diketahui bahwa penentuan jumlah klaster sangat penting dalam algoritma k-means(Ma, Gu, Li, Ma, & Wang, 2015). Beberapa artikel (Barakbah & Kiyoki, 2009; Ma et al., 2015; Yadav & Sharma, 2012) menyatakan algoritma k-means sangat tergantung pada penentuan titik pusat klaster awalnya. Untuk setiap percobaan, dengan pemilihan pusat klaster awal secara acak, algoritma k-means cenderung menghasilkan klaster yang berbeda. Sehingga untuk memastikan klaster terbaik harus dilakukan beberapa kali percobaan, yaitu dengan memperhatikan nilai SSE (*Sum Sequare Error*) yang terkecil. Namun sangat sulit menentukan batasan percobaan agar k-means mendapatkan hasil yang baik. Keadaan dimana algoritma k-means tidak dapat menemukan hasil klaster terbaik diistilahkan dengan terjebaknya k-means pada solusi lokal optima (Barakbah & Kiyoki, 2009). Lokal optima merupakan suatu kondisi dimana klaster telah terjadi atau perulangan telah berhenti, padahal pencarian baru dilakukan untuk sebagian kecil ruang data. Hal tersebut disebabkan oleh titik pusat klaster yang sangat berdekatan sehingga menyebabkan karakteristik klaster yang sangat mirip. Disamping itu penentuan titik pusat klaster awal secara acak dapat menyebabkan suatu data *outlier* terpilih menjadi titik pusat yang menyebabkan kualitas klater menjadi tidak bagus.

Penentuan titik pusat klaster awal menjadi fase yang sangat menentukan dalam setiap penelitian yang melibatkan algoritma k-means. Algoritma Pillar sangat efektif untuk menentukan posisi titik pusat klaster awal dan meningkatkan akurasi hasil pengelompokan(Barakbah & Kiyoki, 2009; Bhusare & Bansode, 2014). Algoritma pillar menentukan titik pusat klaster dengan memilih data yang memiliki nilai *euclidean* paling jauh dari titik pusat klaster dengan tetap memperhatikan kemungkinan data *outlier.* Sehingga selain optimal, algoritma ini juga dapat menghindari kemungkinan data *outlier* menjadi titik pusat klaster. Kualitas klaster juga dapat diperoleh dari penerapan algoritma klaster dinamis. Prinsip kerja dari algoritma k-means klaster dinamis adalah dengan menyusun k buah titik pusat klaster (*centroid*) dari sekumpulan data, kemudian secara berulang-ulang partisi klaster ini diperbaiki dengan memperhatikan nilai PC (*Partition Coefficient)* dan VC (*Variance Cluster*), hingga tidak terjadi perubahan yang signifikan pada partisi klaster. K-Means dengan algoritma klaster dinamis terbukti dapat meningkatkan akurasi model yang terbentuk(Shafeeq B M & K S, 2012), namun memiliki masalah pada jumlah klaster yang berubah tergantung inisialisasi titik pusat klaster awal secara acak.

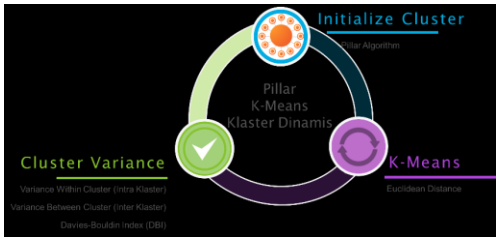
Kesederhanaan proses perhitungan algoritma k-means membuat algoritma ini banyak digunakan dalam penanganan masalah pengelompokan data. Diluar dari ketidaksempurnaan algoritma ini, masalah penentuan titik pusat klaster dan jumlah klaster terbaik selalu menjadi fokus penelitian. Inisialisasi titik pusat klaster dengan algoritma *mean-based* pada algoritma k-means klaster dinamis terbukti meningkatkan kualitas klaster dilihat dari nilai PC, SSE, dan VC(Seputra, Sudarma, & Jasa, 2017). *Mean based* bertugas menentukan titik pusat klaster awal, sedangkan klaster dinamis bertugas menentukan jumlah klaster terbaik. Terdapat tiga proses utama dalam optimasi algoritma *k-means* klaster dinamis, yakni inisialisasi titik pusat klaster awal, selanjutnya titik pusat klaster tersebut menjadi klaster awal dalam perhitungan *k-means*. Proses ketiga adalah perhitungan kualitas klaster dengan membandingkan nilai *cluster variance* dengan *cluster variance* jumlah klaster sebelumnya(Seputra et al., 2017). Penentuan titik pusat klaster awal secara *mean based* dari anggota klaster yang terbentuk berdasarkan jumlah klaster yang dipilih menyebabkan *k-means* melakukan pencarian calon titik pusat klaster baru disekitar titik pusat klaster awal yang telah terbentuk(Pratama & Harjoko, 2017). Sehingga memungkinkan terbentuk sebuah klaster yang tidak mencerminkan karakteristik dari klaster tersebut.

Oleh karena begitu pentingnya fase inisialisasi pusat klaster awal, maka pada penelitian ini diusulkan metode optimasi inisialisasi titik pusat klaster awal *k-means* klaster dinamis menggunakan algoritma pillar. Diharapkan dengan diterapkannya algoritma pillar dapat memperbaiki kualitas dan kecepatan proses penentuan klaster pada algoritma k-means klaster dinamis.

2. METODE PENELITIAN

Penelitian ini mengkombinasikan antara algoritma pillar dengan algoritma *k-means* klaster dinamis yang selanjutnya disebut pillar *k-means* klaster dinamis. Secara garis besar dalam satu siklus

pillar *k-means* klaster dinamis terdapat tiga tahap seperti pada gambar 1 yakni tahap pertama untuk penentuan inisialisasi titik pusat klaster dengan algoritma pillar. Inisialisasi klaster yang terbentuk selanjutnya digunakan dalam tahap kedua, yakni perhitungan anggota klaster menggunakan *k-means*. Nilai VC klaster yang terbentuk dibandingkan dengan nilai VC jumlah klaster sebelumnya untuk menentukan jumlah klaster yang terbaik, serta dengan memaksimalkan jarak anggota klaster dengan tetangganya menggunakan DBI (*Davies-Bouldin Index*). Begitu seterusnya hingga ditemukan nilai VC optimal dengan DBI paling kecil.



Gambar 1. Pillar K-means Klaster Dinamis

Gambar 1 telah menjelaskan bagaimana siklus algoritma pillar *k-means* klaster dinamis bekerja hingga menemukan jumlah klaster terbaik. Berikut dijelaskan lebih detail mengenai fase yang terjadi pada algoritma pillar *k-means* klaster dinamis.

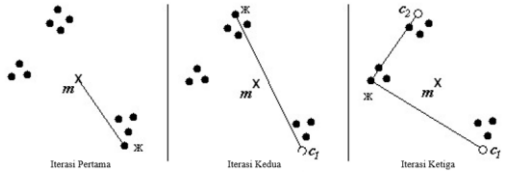
a. Fase inisialisasi Titik Pusat Klaster Awal

Fase inisialisasi merupakan fase awal yang sekaligus sangat menentukan kualitas klaster yang dihasilkan. Oleh karena itu pada tahap ini digunakan algoritma pillar dengan harapan dapat menghasilkan *centroid* awal yang optimal. Algoritma pillar terinspirasi oleh penempatan pilar disebuah bangunan, dimana pilar-pilar harus ditempatkan disetiap sudut bangunan yang terjauh sehingga massa bangunan terkonsentrasi pada setiap pilar. Algoritma ini mampu menemukan *centroid* secara terpisah sejauh mungkin antara *centroid* awal dalam satu distribusi data, serta dapat menghindari terpilihnya data *outlier* sebagai *centroid* awal(Bhusare & Bansode, 2014) seperti pada gambar 2. *Centroid* awal yang dihasilkan pada tahap ini, akan digunakan untuk proses selanjutnya yakni pengelompokan data oleh *k-means*. Berikut dapat dipaparkan secara rinci proses yang terjadi pada tahap inisialisasi dimana X = $\{x_i \mid i=1,..,n\}$ sebagai data, k adalah jumlah klaster, C=$\{c_i \mid i=1,\ldots,k\}$ sebaga *centroid* awal, SX $\subseteq$ X untuk mengidentifikasi data yang dipilih untuk mendai kandidat *centroid,* DM =$\{ x_i \mid i=1,..,n\}$ sebagai akumulasi jarak data, D = $\{x_i \mid i=1,..,n\}$ sebagai jarak data dengan pusat klaster awal, dan m merupakan mean dari data.

1. Set k=jumlah klaster
2. Tentukan *mean* data (m) sebagai titik pusat klaster awal.
3. Tentukan i=1 sebagai penghitung iterasi kandidat *centroid.*
4. Hitung jarak data dengan pusat klaster D[n] = dis(X, m) menggunakan persamaan 1

$$dis(X,m) = |\sum_{j=1}^{N} X_j - m_j| \quad (1)$$

5. Hapus data *outlier* D[n]
6. Tentukan Dmax = argmax(D)
7. DM = DM+D
8. Tentukan x = $x_{argmax(DM)}$ sebagai kandidat *centroid* ke i
9. SX = SX ∪ x
10. Tentukan C[i]=D(SX)
11. m=mean (C)
12. i=i+1
13. jika i<k, kembali ke langkah 4
14. Selesai dengan C sebagai centroid awal



Gambar 2. Ilustrasi Algoritma Pillar

Dalam beberapa penelitian sering kali ditemukan suatu data yang nilainya jauh berbeda dengan sebagian besar nilai lain dalam kelompoknya yang disebut *outlier.* Penghapusan *outlier* menjadi sangat penting untuk meningkatkan kualitas data serta meningkatkan efisiensi perhitungan (Sunitha, Balraju, Sasikiran, & Ramana, 2014). Oleh karena itu pada penelitian ini data *outlier* dihapus sebelum tahap penentuan calon *centroid* meenggunakan metode *Interquartile Range* (IQR). IQR bekerja dengan membagi data (D) menjadi empat bagian yakni Q_1 = data dari nilai minimal hingga median, Q_2 = median data, Q_3 = dari dari median hingga nilai paling tinggi. Adapun persamaan 2 yang digunakan untuk mendapatkan IQR.

$$IQR = Q3\text{-}Q1 \quad (2)$$

$$Lower\ Limit = Q_1 - (1.5 * IQR) \quad (3)$$

$$Upper\ Limit = Q_3 + (1.5 * IQR) \quad (4)$$

Setelah diperolehnya nilai IQR maka dicari nilai ambang batas bawah data (*Lower Limit*) menggunakan persamaan 3 dan nilai ambang batas atas (*Upper Limit*) dengan persamaan 4. Data D[n] dapat dikatakan *outlier* jika *Lower Limit* > D[n] > *Upper Limit*, maka D[n] dihapus. Setelah *outlier* terhapus, maka dilanjutkan dengan penentuan nilai *distance* (D) paling tinggi untuk dijadikan kandidat *centroid.*

b. Fase Pembentukan Kelompok Data

Centroid awal yang dihasilkan oleh algoritma pillar digunakan pada proses *clustering* menggunakan algoritma *k-means.* Adapun tahapannya sebagai berikut.

1. Menghitung jarak data (x) dengan *centroid* menggunakan *euclidean* (m) sesuai persamaan 5.

$$d(X,m) = \sum_{i=1}^{k}\sum_{y=1}^{n}\sqrt{\sum_{j=1}^{q}(X_{yj} - m_{ij})^2} \quad (5)$$

2. Kelompokan data kedalam klaster dengan jarak minimal.
3. Hitung kembali nilai pusat klaster dengan menghitung nilai rata-rata pada setiap klaster dengan persamaan 6.

$$c_i = \sum_{j=1}^{n_{si}} m_{ij} \in s_i \quad (6)$$

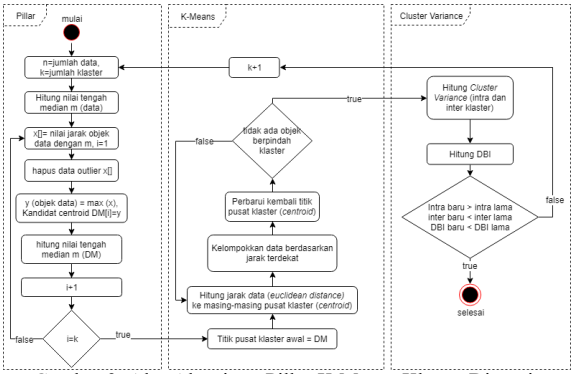
4. Hitung kembali klaster masing-masing data hingga anggota klaster tidak berubah

c. Fase Pengujian

Untuk menentukan jumlah klaster yang paling optimal dan dapat memvalidasi apakah partisi yang diterapkan dalam proses *clustering* sesuai dengan data, digunakan indeks pengukuran validitas klaster. Adapun metode yang digunakan yakni

1. *Partition Coefficient* (PC) merupakan metode yang mengukur jumlah *cluster* yang mengalami *overlap.* Jumlah klaster paling optimal dapat ditentukan dari nilai PC yang paling besar.
2. *Sum Squared Error* (SSE) merupakan metode yang mengukur jumlah *square error* pada setiap klaster. Nilai klaster optimal ditentukan dari nilai SSE yang paling kecil.
3. *Cluster Variance* digunakan untuk mengetahui penyebaran dari data hasil *clustering.* Nilai klaster optimal dapat dilihat dari nilai *Variance* yang semakin kecil. Ada dua macam *cluster variance* yaitu *variance within cluster* (intra klaster) dan *variance between cluster* (inter klaster). Istilah intra digunakan untuk mengukur kekompakan dari suatu kelompok. Sedangkan inter adalah minimum jarak antar pusat klaster. Inter digunakan untuk mengukur pemisahan antar klaster (Bunkers, Miller, & DeGaetano, 1996).
4. *Davies-Bouldin Index* (DBI)
   DBI adalah rasio antara jumlah *cluster scatter* sampai dengan *cluster sparation*(Maulik & Bandyopadhyay, 2002). DBI adalah dengan memaksimalkan jarak inter klaster dan meminimalkan jarak intra klaster. Nilai DBI minimum merupakan skema *clustering* terbaik (Bala, Basu, & Dasgupta, 2015).

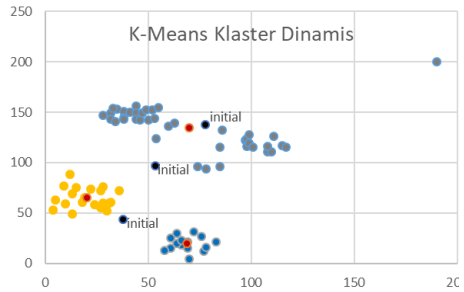
Lebih jelas mengenai alur kerja algoritma pillar *k-means* klaster dinamis dapat dilihat pada gambar 3.



Gambar 3. Alur Algoritma Pillar K-Means Klaster Dinamis

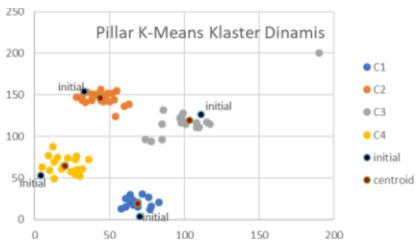
3. HASIL DAN PEMBAHASAN

Pengujian pertama dilakukan terhadap data set ruspini (Segneri, 2017). Data dua dimensi dipilih sebagai data uji untuk mengetahui gambaran data yang dihasilkan oleh algoritma pillar pada tahap inisialisasi *centroid* awal. Adapun pengujian menggunakan aplikasi berbasis website yang dikembangkan menggunakan *framework php codeigniter* dengan *MySQL* sebagai basis data.



Gambar 4. Penentuan *Centroid* Awal *K-means* Klaster Dinamis

Hasil pengujian pertama memperlihatkan perbedaan bentuk inisialisasi *centroid* yang dihasilkan kedua algoritma. Perbedaan yang mencolok antara kedua algoritma tersebut terlihat dalam hal penentuan inisialisasi *centroid* awal, dimana *k-means* klaster dinamis menghasilkan inisialisasi *centroid* berbentuk linear berwarna biru gelap seperti terlihat pada gambar 4.



Gambar 5. Penentuan *Centroid* Awal *Pillar K-means* Klaster Dinamis

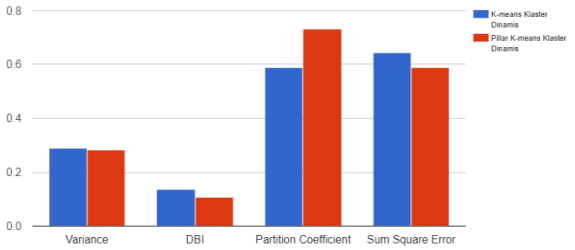
Sedangkan *pillar k-means* klaster dinamis menghasilkan inisialisasi *centroid* berbentuk pilar berwarna biru gelap sesuai dengan yang diharapkan seperti terlihat pada gambar 5. Tahap evaluasi merupakan tahap pengujian terhadap hasil clustering. Pengujian dilakukan untuk melihat

apakah kombinasi algoritma *k-means* klaster dinamis dengan algoritma *pillar* sebagai inisialisasi klaster awal mampu menghasilkan kualitas klaster yang lebih baik dibandingkan dengan *k-means* klaster dinamis. Pengujian dilakukan dengan mengukur tingkat validitas masing-masing algoritma menggunakan metode *Cluster Variance* (V), *Davies Bound Index* (DBI), *Partition Coefficient* (PC), dan *Sum Squared Error* (SSE) seperti terlihat pada tabel 1.

Tabel 1. Hasil Pengujian Data Set Ruspini

Algorithm	K	VW	VB	V	DBI	PC	SSE
***K-means* Klaster Dinamis**	1	32.50	1.00	32.50	0.43	1.98	7.50
***K-means* Klaster Dinamis**	2	11.82	41.69	0.28	0.20	3.95	5.81
***K-means* Klaster Dinamis**	3	9.82	33.99	0.29	0.14	5.89	6.44
Pillar *K-means* Klaster Dinamis	1	32.50	1.00	32.50	0.43	0.97	9.72
Pillar *K-means* Klaster Dinamis	2	11.82	41.69	0.28	0.20	3.43	5.93
Pillar *K-means* Klaster Dinamis	3	11.37	35.59	0.32	0.57	6.55	5.78
Pillar *K-means* Klaster Dinamis	4	9.98	35.21	0.28	0.11	7.30	5.88

Pengujian dilakukan dengan menetapkan satu buah klaster awal sebagai inisialisasi sekaligus sebagai klaster pembanding untuk menentukan kualitas klaster berikutnya. Jumlah klaster akan bertambah sesuai dengan hasil perbandingan nilai validitas klaster dengan klaster sebelumnya seperti pada tabel 1. Dari hasil pengujian menggunakan ruspini data set (Segneri, 2017) seperti pada tabel 1, terdapat perbedaan jumlah klaster yang dihasilkan pada akhir iterasi. Algoritma *pillar k-means* klaster dinamis menghasilkan 4 klaster, sementara algoritma *k-means* klaster dinamis menghasilkan 3 klaster. Jumlah klaster optimal dipilih dari jumlah klaster terakhir dari masing-masing algoritma yakni jumlah klaster 4 pada algoritma *pillar k-means* klaster dinamis dan jumlah 3 klaster pada algoritma kmeans klaster dinamis. Jika dilihat dari nilai V, DBI, SSE algoritma *pillar k-means* klaster dinamis pada 4 klaster yang lebih kecil dan PC lebih besar dibandingkan dengan *k*-means klaster dinamis dengan 3 klaster seperti dilihat pada gambar 6, maka penerapan algoritma *pillar* dalam menentukan inisialisasi *centroid* awal terbukti meningkatkan kinerja algoritma *k-means* klaster dinamis. Sehingga jumlah 4 klaster yang dihasilkan oleh algoritma *pillar k-means* klaster dinamis dipilih sebagai klaster terbaik. Selain disebabkan oleh penghapusan data *outlier* pada algoritma *pillar k-means* klaster dinamis, perbedaan jumlah klaster yang dihasilkan juga disebabkan oleh perbedaan metode dalam penentuan kondisi klaster dinamis. Dalam algoritma *pillar k-means* klaster dinamis k+1 ditentukan dengan meminimalkan nilai inter klaster (VB) dan DBI dari jumlah klaster sebelumnya. Sedangkan pada algoritma *k-means* klaster dinamis hanya dengan meminimalkan nilai VB.



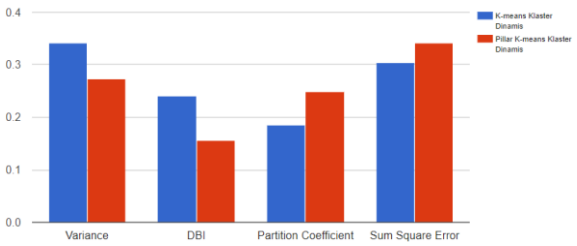
Gambar 6. Perbandingan Hasil Pengujian Ruspini Data Set

Pengujian juga dilakukan terhadap Iris Data set (Fisher, 1993) untuk membuktikan apakah algoritma pillar *k-means* klaster dinamis bekerja dengan baik pada data *multivariate*. Hasil pengujian seperti pada tabel 2.

Tabel 2. Hasil Pengujian Iris Data set

Algorithm	K	VW	VB	V	DBI	PC	SSE
***K-means* Klaster Dinamis**	1	0.78	1.00	0.78	0.23	0.09	0.32
***K-means* Klaster Dinamis**	2	0.34	0.99	0.34	0.24	0.18	0.30
Pillar *K-means*Klaster Dinamis	1	0.78	1.00	0.78	0.23	0.13	0.40
Pillar *K-means*Klaster Dinamis	2	0.34	0.99	0.34	0.24	0.19	0.33
Pillar *K-means*Klaster Dinamis	3	0.26	0.95	0.27	0.16	0.25	0.34

Dilihat dari tabel 2, menunjukkan bahwa kedua algoritma menghasilkan jumlah klaster yang berbeda pada akhir iterasi, dimana k-means klaster dinamis menghasilkan 2 klaster, sementara *pillar k-means* klaster dinamis dengan 3 klaster. Nilai SSE pada jumlah klaster 3 algoritma *pillar k-means* klaster dinamis memang lebih tinggi dari algoritma *k-means* klaster dinamis. Namun pada nilai validitas klaster lainnnya seperti DBI, dan V, *pillar k-means* klaster dinamis terbukti memiliki penyebaran dengan kedekatan anggota klaster yang lebih baik karena memiliki nilai DBI dan V lebih kecil. Serta nilai PC yang lebih besar menandakan bahwa kualitas anggota klaster yang dihasilkan dari algoritma *pillar k-means* klaster dinamis lebih baik.



Gambar 7. Perbandingan Hasil Pengujian Iris Data Set

4. KESIMPULAN

Berdasarkan hasil pengujian terhadap ruspini data set diperoleh hasil dengan nilai V, DBI, dan SSE pada akhir iterasi algoritma *pillar k-means* klaster dinamis memperoleh nilai lebih kecil masing-masing 0.28, 0.11, dan 5.88 dibandingkan dengan nilai V, DBI, dan SSE dari algoritma *k-means* klaster dinamis masing- masing sebesar 0.29, 0.14, dan 6.44. Serta dengan nilai PC sebesar 7.3 lebih besar dari nilai PC algoritma k-means klaster dinamis telah membuktikan bahwa penerapan inisialisasi klaster menggunakan algoritma *pillar* telah meningkatkan kualitas klaster yang dihasilkan dari algoritma *k-means* klaster dinamis khususnya pada pengujian ruspini data set. Kinerja baik dari algoritma *pillar k-means* klaster dinamis juga sangat dipengaruhi oleh penghapusan *outlier* pada proses inisialiasai *centroid* awal, sehingga kemungkinan terpilihnya *outlier* menjadi pusat klaster dapat dihindari dan perhitungan menjadi lebih efisien. Namun terdapat sedikit perbedaan hasil pengujian terhadap *iris* data set, dimana nilai SSE algoritma pillar *k-means* klaster dinamis sebesar 0.34 lebih besar dari nilai SSE algoritma *k-means* klaster dinamis sbesar 0.32. Hal ini dapat disebabkan oleh kurang akuratnya penentuan data *outlier* pada *iris* data set yang bersifat multivariat, sehingga memungkinkan data outlier menjadi *centroid* awal klaster. Sehingga jika dilihat dari nilai validitas SSE, algoritma *pillar k-means* klaster dinamis masih kurang bekerja optimal dibandingkan dengan algoritma *k-means* klaster dinamis. Untuk penelitian selanjutnya, agar mengoptimalkan perhitungan data outlier, sehingga algoritma *pillar* mampu diterapkan dengan baik pada data multivariat.

DAFTAR PUSTAKA

AGGARWAL, N., AGGARWAL, K., & GUPTA, K. (2012). Comparative Analysis of K-means and Enhanced K-means Clustering Algorithm for Data Mining. *International Jouranl of Scientific & Engineering Research*, *3*(3).

BALA, C., BASU, T., & DASGUPTA, A. (2015). Automatic detection of k with suitable seed values for classic k-means algorithm using de. *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*, 759–765. https://doi.org/10.1109/ICACCI.2015.7275702

BARAKBAH, A. R., & KIYOKI, Y. (2009). A pillar algorithm for k-means optimization by distance maximization for initial centroid designation. *2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 - Proceedings*, 61–68. https://doi.org/10.1109/CIDM.2009.4938630

BHUSARE, B. B., & BANSODE, S. M. (2014). *Centroids Initialization for K-Means Clustering using Improved Pillar Algorithm*. *3*(4), 1317–1322.

BUNKERS, M. J., MILLER, J. R., & DEGAETANO, A. T. (1996). Definition of climate regions in the northern plains using an objective cluster modification technique. *Journal of Climate*, Vol. 9, pp. 130–146. https://doi.org/10.1175/1520-0442(1996)009<0130:DOCRIT>2.0.CO;2

FISHER, R. . (1993). Iris Data Set. Tersedia di:<https://archive.ics.uci.edu/ml/datasets/iris> [Diakses 22 November 2019]

MA, L., GU, L., LI, B., MA, Y., & WANG, J. (2015). An Improved K-means Algorithm based on Mapreduce and Grid. *International Journal of Grid and Distributed Computing*, *8*(1), 189–200. https://doi.org/10.14257/ijgdc.2015.8.1.18

MAULIK, U., & BANDYOPADHYAY, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(12), 1650–1654. https://doi.org/10.1109/TPAMI.2002.1114856

PRATAMA, I. P. A., & HARJOKO, A. (2017). Penerapan Algoritma Invasive Weed Optimnization untuk Penentuan Titik Pusat Klaster pada K-Means. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *9*(1), 65. https://doi.org/10.22146/ijccs.6641

SEGNERI, C. (2017). K-Means Clustering. Tersedia di:<https://rpubs.com/cjsegneri/kmeansclustering> [Diakses 22 November 2019]

SEPUTRA, K. A., SUDARMA, I. M., & JASA, L. (2017). The Optimization of the Dynamic K-Means Clustering Algorithm with the Cluster Initialization in Grouping Travelers Perception to the Beach Tourist Destinations in Bali , Indonesia. *International Journal of Research in IT, Management and Engineering, ISSN 2249-1619*, *07*(04), 1–7.

SHAFEEQ B M, A., & K S, H. (2012). Dynamic Clustering of Data with Modified K-Means Algorithm. *International Conference on Information and Computer Networks (ICICN 2012)*, *27*(Icicn), 221–225. https://doi.org/10.13140/2.1.4972.3840

SUNITHA, L., BALRAJU, M., SASIKIRAN, J., & RAMANA, E. V. (2014). *Automatic Outlier Identification in Data Mining Using IQR in Real-Time Data*. *3*(6), 7255–7257.

WU, J. (2012). *Advances in K-means Clustering*. 1–16. https://doi.org/10.1007/978-3-642-29807-3

YADAV, R., & SHARMA, A. (2012). Advanced methods to improve performance of k-means algorithm: A review. *Global Journal of Computer Science and Technology*, *12*(9), 47–51.

Halaman ini sengaja dikosongkan