

KLASTERING DATA MENGGUNAKAN ALGORITMA DYNAMIC K-MEANS

Widiarina

Abstract— *The disadvantage of the K-means algorithm is sensitive to have problems determining the initial partition number of clusters (k) determining the initial value that is different may produce different cluster groups. To solve the problem of the sensitivity of the initial partition number of clusters in K-means algorithm, the proposed algorithm dynamic cluster. The result showed that the Dynamic K-means algorithm, can produce quality cluster that is more optimal than the K-means.*

Intisari— Salah satu kekurangan algoritma *K-means* yaitu mempunyai masalah sensitif terhadap penentuan partisi awal jumlah *cluster(k)* penentuan nilai awal yang berbeda mungkin dapat menghasilkan kelompok *cluster* yang berbeda pula. Untuk menyelesaikan masalah sensitifitas partisi awal jumlah *cluster* pada algoritma *K-means*, maka diusulkan algoritma *cluster* dinamik. Hasil percobaan menunjukkan bahwa algoritma *Dynamic K-means*, dapat menghasilkan kualitas *cluster* yang lebih optimal dibandingkan dengan *K-means*.

Kata kunci : Segmentasi Pelanggan, K-Means, quality cluster

I. PENDAHULUAN

Algoritma *K-means* adalah algoritma *clustering* yang paling populer digunakan karena memiliki kelebihan yaitu algoritma yang sederhana dan mudah diimplementasikan. Salah satu kekurangan algoritma *K-means* yaitu mempunyai masalah sensitif terhadap penentuan partisi awal jumlah *cluster(k)* dan solusi akhir menyatu pada *local minima*. Penentuan partisi jumlah *cluster(k)* sangat penting bagi algoritma *K-means*, tetapi tidak ada ketentuan yang berlaku untuk menentukan berapa jumlah *cluster(k)* yang akan dibentuk penentuan partisi jumlah *cluster* diawal yang sangat sulit, penentuan nilai awal yang berbeda mungkin dapat menghasilkan kelompok *cluster* yang berbeda pula.

Beberapa penelitian yang telah dilakukan mengenai masalah sensitifitas inisialisasi jumlah *cluster(k)*, dan algoritma yang digunakan:

Deelers dan Auwatanamongkol, dalam paper yang berjudul “*Enhancing K-means Algorithm With Initial Cluster Center Derived from Data Partitioning along the Data Axis with the Highest Variance*”, algoritma yang digunakan adalah algoritma partisi data. Baolin Yi dalam paper yang berjudul “*An Improved Initialization Center Algorithm for K-means Clustering*”, algoritma yang digunakan adalah algoritma inisialisasi pusat cluster berbasis kepadatan (*density*). Zhang dan Fang, dalam judul “*An improved K-means Algorithm*”, metode yang digunakan Genetic algoritim (GA).

Berdasarkan penelitian-penelitian terdahulu, metode *K-means* paling banyak digunakan dalam segmentasi, untuk memecahkan masalah sensitifitas partisi awal jumlah *cluster(k)*, pada penelitian ini mengusulkan algoritma *cluster* dinamik pada algoritma *K-means* untuk menghasilkan kualitas *cluster* yang optimal sehingga dapat memberikan hasil yang lebih baik dan tepat.

Maksud dari penelitian ini adalah untuk menerapkan algoritma *dynamic K-means* untuk mendapatkan kualitas *cluster* yang optimal. Hasil *cluster* akan di evaluasi menggunakan Davies-Bouldin Index (DBI).

II. KAJIAN LITERATUR

- a. Algoritma Segmentasi

“Tujuan dari algoritma segmentasi adalah dengan memecahkan setiap data dalam dataset menjadi kelompok-kelompok yang homogen. Kelompok data ini biasanya disebut sebagai segmen atau *cluster*. Setiap segmen yang terbentuk akan terdiri dari data yang sejenis dan berbeda dengan data pada segmen lainnya”[8].

Metode segmentasi dapat diklasifikasikan menjadi beberapa jenis berdasarkan logika segmentasi[8]:

 1. Metode partisi
 2. Baik dalam mengelompokkan data menjadi kelompok yang sudah ditentukan lebih dahulu
 3. Metode hierarki
 4. Memecah data kedalam segmen dengan struktur hirarki. Data setiap segmen tetap homogen, namun memiliki tingkatan antara satu segmen dengan segmen lainnya
 5. Metode berbasis kepadatan
 6. Merupakan perpaduan kedua metode sebelumnya, metode ini memecah data kedalam partisi berdasarkan jarak data terhadap setiap segmen. Namun setiap segmen memiliki batasan jarak, sehingga nilai tidak boleh lebih kecil dari nilai minimum segmen.

Program Studi Manajemen Informatika AMIK Bina Sarana Informatika Bekasi, Jln. Cut Mutia No. 88 Bekasi Timur Telp. (021) 82425638, e-mail: widiarina.wda@bsi.ac.id

7. Metode grid

Segmen terbentuk bersarakan sturktur ruangan yang seperti sel. Metode ini dapat mengolah data besar dengan cepat, namun memiliki akurasi yang rendah.

Algoritma *K-means* termasuk kedalam algoritma segmentasi dengan metode partisi. Selain algoritma *K-means*, algoritma segmentasi dengan metode partisi lainnya yaitu algoritma *Partitioning Around Medoids (PAM)*, *K-Medoids*, dan *Nearest Neighbor clustering*.

b. Algoritma *K-means*

"*K-means* adalah algoritma untuk pelatihan *unsupervised*, pertama kali dipublikasikan oleh Stuart Lloyd pada tahun 1984 dan merupakan algoritma *clustering* yang banyak digunakan. *K-means* merupakan suatu algoritma pengklasteran yang cukup sederhana yang mempartisi dataset kedalam beberapa kluster k. Algoritmanya cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan"[10].

Prinsip utama dari teknik ini adalah menyusun k buah partisi/pusat massa (*centroid*)/rata-rata (*mean*) dari sekumpulan data. Algoritma *K-means* dimulai dengan pembentukan partisi kluster di awal kemudian secara iteratif partisi kluster ini diperbaiki hingga tidak terjadi perubahan yang signifikan pada partisi kluster [9].

Algoritma *K-means* menggunakan pendekatan "top-down" dari awal sudah ditetapkan jumlah kluster kemudian dilakukan pengamatan. Pendekatan ini menghitung dengan cepat dan dapat menangani jumlah data yang besar. Dalam algoritma pengelompokan, sebuah data dikatakan serupa dengan mengukur jarak nilai dari satu data dengan data lain [7].

Proses algoritma *K-means* dapat diurutkan sebagai berikut: Pertama, jumlah K dari observasi dipilih secara acak diantara semua jumlah N pengamatan sesuai dengan jumlah *cluster*, yang akan menjadi pusat *cluster* awal. Kedua, masing-masing dari sisa $N - K$ pengamatan, mencari *cluster* terdekat berdasarkan jarak *Euclidean*, setelah masing-masing diberikan *cluster* terdekat, *recompute* pusat *cluster*. Terakhir, setelah mengalokasikan semua pengamatan, hitung jarak *Euclidean* antara masing-masing pengamatan dan titik pusat *cluster* dan mengkonfirmasi apakah akan dialokasikan ke pusat *cluster* atau tidak. Berikut tahapan Algoritma *K-means*:

Tahap 1:

Membuat partisi sejumlah k dari segmentasi yang akan dibentuk

Tahap 2:

Mengisi setiap obyek dalam dataset kedalam segmen terdekat

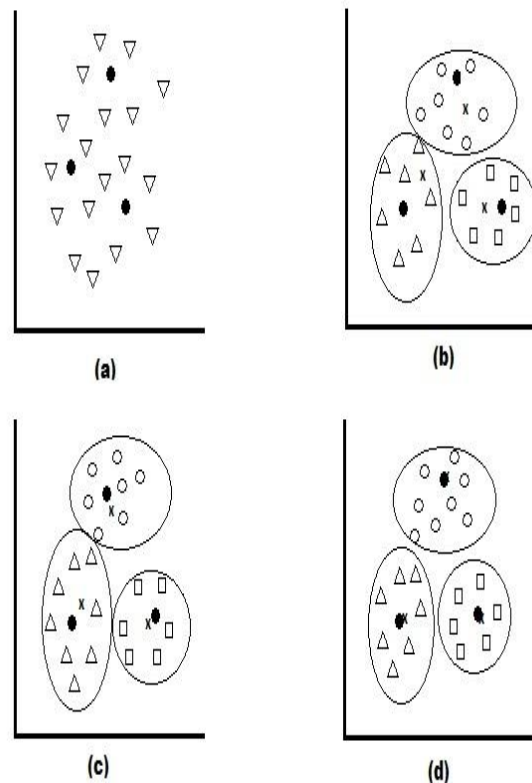
Tahap 3:

Kalkulasi ulang setiap segmentasi yang terbentuk

Tahap 4:

Ulangi langkah 2 dan 3 hingga data di dalam segmentasi tidak berubah

Gambar 1 mengilustrasikan penerapan algoritma *K-means* untuk set pengamatan pada bidang *euclidean*, misalkan ditetapkan $K=3$. Gambar 1 menggambarkan pengamatan dua dimensi, titik akan dibagi kedalam tiga *cluster*. Pada fase inisialisasi, dipilih tiga pengamatan secara sembarang, pada gambar ditunjukkan dengan titik-titik besar yang merupakan tiga *centroid* awal. Pada iterasi pertama, setiap objek diberikan ke *centroid* yang terdekat, hasil *cluster* ditunjukkan pada Gambar 1(b) dan *centroid* ditunjukkan dengan simbol x. Gambar 1(b) menunjukkan *centroid* baru, dihitung sebagai rata-rata atribut untuk objek milik masing-masing *cluster* dan direpresentasikan sebagai titik besar. Gambar 1(c) dan (d) masing-masing menunjukkan iterasi kedua dan ketiga dari algoritma, karena pada akhir iterasi ketiga objek tidak berubah, maka algoritma berhenti.



Sumber: Hasil Penelitian (2015)

Gambar 1. Contoh aplikasi algoritma *K-means*

Contoh penerapan algoritma *K-Means*:

1. Membuat partisi sejumlah k dari segmentasi yang akan dibentuk

$$M = [m_1, \dots, m_k]; \dots\dots\dots (1)$$

2. Mengisi setiap obyek dalam dataset kedalam segmen terdekat

$$x_j \in C_i, \text{ if } \|x_j - m_i\| < \|x_j - m_l\| \text{ for } j = 1, \dots, N, i \neq l, \text{ and } i = 1, \dots, K; \dots\dots\dots(2)$$

Dengan jarak *Euclidean* untuk menghitung jarak antara dua objek:

$$d = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \dots\dots\dots (3)$$

Jarak *Jaccard* untuk mengukur jarak data yang biner:

$$d = \frac{Count_{10} + Count_{01}}{Count_{11} + Count_{10} + Count_{01}} \dots\dots(4)$$

3. Kalkulasi ulang setiap segmentasi yang terbentuk

$$m_l = \frac{1}{N_l} \sum_{x_j \in C_l} x_j \dots\dots\dots(5)$$

4. Ulangi langkah 2 dan 3 hingga data di dalam segmentasi tidak berubah

Contoh pengklasteran menggunakan *k-means*, contoh data pada Tabel 1.

Tabel 1. Data yang akan disegmentasi

Name	Variabel 1	Variabel 2	Variabel 3	Variabel 4	Variabel 5
A	7	8	4	5	2
B	6	8	5	4	2
C	8	9	7	8	9
D	6	7	7	7	8
E	1	2	5	3	4
F	3	4	5	3	5
G	7	8	8	6	6
H	8	9	6	5	5
I	2	3	5	6	5
J	1	2	4	4	2
K	3	2	6	5	7
L	2	5	6	8	9
M	3	5	4	6	3
N	3	5	5	6	3

Sumber: Hasil Penelitian (2015)

Tahap 1 : k = 3

Tahap 2 :

Data secara acak akan dipilih sebagai data awal dalam membentuk segmentasi, contoh D, K dan M seperti terlihat pada Gambar 2.

	Klaster1	Klaster2	Klaster3
Step 1	D	K	M

Sumber: Hasil Penelitian (2015)

Gambar 2. Data acak sebagai data awal segmen

Setelah itu data lainnya akan dikalkulasi berdasarkan jarak *Euclidean* dari data tersebut terhadap data yang dipilih. Setelah perhitungan ke setiap titik *Euclidean* dilakukan, maka data akan mengikuti segmentasi dengan titik *Euclidean* yang terdekat.

Contoh perhitungan jarak *euclidean* dari A ke D

$$d = \sqrt{(7-6)^2 + (8-7)^2 + (4-7)^2 + (5+7)^2 + (2-8)^2}$$

$$= \sqrt{1 + 1 + 9 + 4 + 36}$$

$$= \sqrt{51} = 7.1$$

Perhitungan yang sama dilakukan untuk semua titik kedalam semua segmentasi hingga didapat data yang tersebar seperti Gambar 3 Untuk menyempurnakan data dalam segmentasi, dihitunglah nilai titik tengah untuk setiap segmentasi dengan merata-rata variable dalam setiap segmentasi seperti pada Tabel 2.

	Klaster1	Klaster2	Klaster3
Step 2	C	E	A
	G	F	B
	D	K	M
	L	I	N
	H		J

Sumber: Hasil Penelitian (2015)

Gambar 3. Kumpulan data awal untuk setiap segmen

Tabel 2. Nilai titik tengah setiap segmentasi

Segmen 1					
Name	Variabel 1	Variabel 2	Variabel 3	Variabel 4	Variabel 5
C	8	9	7	8	9
D	6	7	7	7	8
G	7	8	8	6	6
H	8	9	6	5	5
L	2	5	6	8	9
Center	6.2	7.6	6.8	6.8	7.4
Segmen 2					
Name	Variabel 1	Variabel 2	Variabel 3	Variabel 4	Variabel 5
E	1	2	5	3	4
F	3	4	5	3	5
I	2	3	5	6	5
K	3	2	6	5	7
Center	2.25	2.75	5.25	4.25	5.25
Segmen 3					
Name	Variabel 1	Variabel 2	Variabel 3	Variabel 4	Variabel 5
A	7	8	4	5	2
B	6	8	5	4	2
J	1	2	4	4	2
M	3	5	4	6	3
N	3	5	5	6	3
Center	4	5.6	4.4	5	2.4

Sumber: Hasil Penelitian (2015)

Tahap 3 :

Jarak *Euclidian* diukur kembali dari setiap titik ke nilai tengah segmentasi yang didapat. Data akan dipindah ke segmentasi dengan nilai *Euclidian* yang terdekat.

Tahap 4 :

Proses ini diulangi sampai tidak ada lagi data yang berpindah dari satu segmentasi ke segmentasi lainnya.

Tabel 2 menggambarkan data setelah pusat segmen baru dihitung, algoritma akan berulang kembali dengan menghitung *euclidean distance* untuk setiap data terhadap pusat segmen yang baru. Namun dalam perhitungan yang dilakukan, tidak ada data yang berpindah kedalam segmen lainnya. Karena terdiri dari data yang sama, pusat segmen tidak akan bergeser dan perhitungan tidak dilanjutkan.

c. Davies-Bouldin Index (DBI)

“*Davies-Bouldin Index* adalah fungsi rasio dari jumlah antara *cluster scatter* sampai dengan *cluster separation*” [6].

Davies-Bouldin Index merupakan metode validasi *cluster* dari hasil *clustering*. Pendekatan pengukuran DBI yaitu memaksimalkan jarak *inter cluster* serta meminimalkan jarak *intra cluster*.

Jarak *intra cluster* dihitung dengan:

$$S_i = \frac{1}{|C_i|} \sum_{x \in C_i} \{\|x - z_i\|\} \dots\dots\dots (6)$$

Dengan C_i adalah banyaknya titik yang termasuk kedalam *cluster* i , x adalah data, dan Z_i adalah *centroid* dari *cluster* i .

Jarak *inter cluster* didefinisikan dengan:

$$d_{ij} = \|z_i - z_j\| \dots\dots\dots (7)$$

Dengan z_i adalah *centroid cluster* i , dan z_j adalah *centroid cluster* j . Maka *Davies-Bouldin Index* didefinisikan sebagai:

$$DB = \frac{1}{K} \sum_{i=1}^K R_i, qt \dots\dots\dots (8)$$

Dimana $R_i, qt = \max_{j, j \neq i} \left\{ \frac{S_i, q + S_j, q}{d_{ij, t}} \right\}$, dengan K adalah banyaknya *cluster*.

Davies-Bouldin Index mengukur rata-rata kemiripan antara masing-masing *cluster* dan salah satu yang paling mirip. Nilai *DB Index* yang minimum adalah skema *clustering* yang paling optimum.

Davies-Bouldin Index didefinisikan sebagai:

$$DB = \frac{1}{K} \sum_{i=1}^K R_i, qt \dots\dots\dots (9)$$

Dimana $R_i, qt = \max_{j, j \neq i} \left\{ \frac{S_i, q + S_j, q}{d_{ij, t}} \right\}$, dengan K adalah banyaknya *cluster*.

Davies-Bouldin Index mengukur rata-rata kemiripan antara masing-masing *cluster* dan salah satu yang paling mirip. Nilai *DB Index* yang minimum adalah skema *clustering* yang paling optimum.

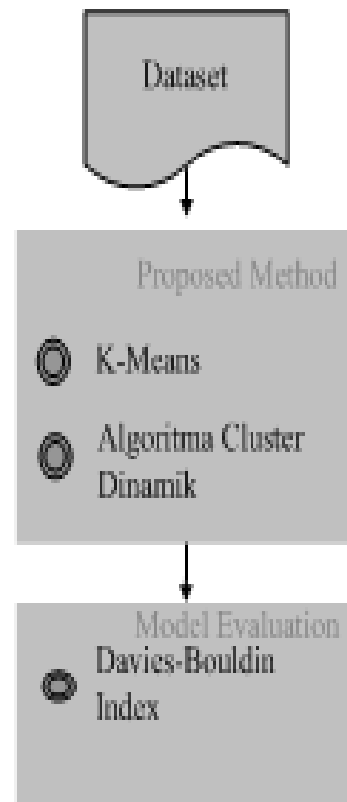
III. METODE PENELITIAN

Penelitian ini menggunakan penelitian eksperimen, yaitu penelitian yang melibatkan penyelidikan perlakuan pada parameter atau variabel tergantung dari penelitiannya dan

menggunakan tes yang dikendalikan oleh si peneliti itu sendiri.

a. Model General

Model yang diusulkan dalam penelitian menerapkan algoritma *cluster* dinamik pada algoritma *K-means* untuk segmentasi data yang terlihat pada Gambar 4.



Sumber: Hasil Penelitian (2015)

Gambar 4. Model yang diusulkan

b. Algoritma Cluster Dynamic

Algoritma *cluster dynamic* cara kerja diawal sama dengan algoritma *k-means*, diakhir jika jarak *intra* lebih kecil dan jika jarak *intra* lebih besar, maka algoritma menghitung *cluster* baru dengan menambahkan *counter* k dengan satu atau $k=k+1$ disetiap iterasi sampai memenuhi batas validitas kualitas *cluster* yang berkualitas.

Istilah *inter* adalah minimum jarak antar pusat *cluster*, *inter* digunakan untuk mengukur pemisahan antar *cluster*, yang didefinisikan sebagai:

$$Inter = \min \{\|m_k - m_{kk}\|\} \quad \forall k = 1, 2, \dots, K - 1 \text{ dan } k = k + 1, \dots, K \dots\dots\dots (10)$$

Istilah *intra* digunakan untuk mengukur kekompakan dari suatu kelompok. Standar deviasi digunakan untuk memeriksa kedekatan titik data setiap *cluster*, dan dihitung sebagai:

$$\sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - X_m)^2} \dots\dots\dots (11)$$

Algoritma *cluster* dinamik adalah sebagai berikut:

1. Membuat partisi sejumlah k dari segmentasi yang akan dibentuk
2. Mengisi setiap obyek dalam dataset kedalam segmen terdekat
3. Kalkulasi ulang setiap segmentasi yang terbentuk
4. Ulangi langkah 2 dan 3 hingga data di dalam segmentasi tidak berubah
5. Hitung jarak inter *cluster*, menggunakan persamaan 3.1
6. Hitung jarak intra *cluster*, menggunakan persamaan 3.2
7. Jika jarak intra *cluster* baru < jarak intra lama dan jarak inter *cluster* baru > jarak inter *cluster* lama, lanjutkan langkah 8, jika tidak ke langkah 9.
8. $k=k+1$, jalankan langkah 1
9. STOP

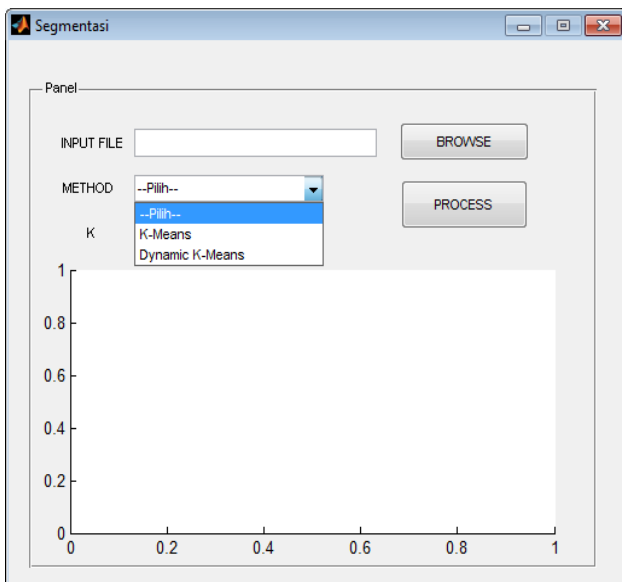
c. Evaluasi dan Hasil

Hasil segmentasi yang terbentuk akan dievaluasi menggunakan *Davies-Bouldin(DB) Index*. Semakin kecil nilai *DB Index* menunjukkan skema *cluster* yang paling optimal.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini, akan digunakan UCI dataset mengenai data pelanggan grosir, data akan diolah menggunakan algoritma *K-means* dan Algoritma *Dynamic K-means*.

Proses pengujian dilakukan dengan program aplikasi yang dibuat menggunakan *software* Matlab yang dapat dilihat pada Gambar 5. Model yang terbentuk akan di evaluasi menggunakan *Davies-Bouldin(DB) Index*. Sebagai ukuran pengujian, data yang sama diterapkan menggunakan algoritma *K-means* dan *Dynamic K-means*.

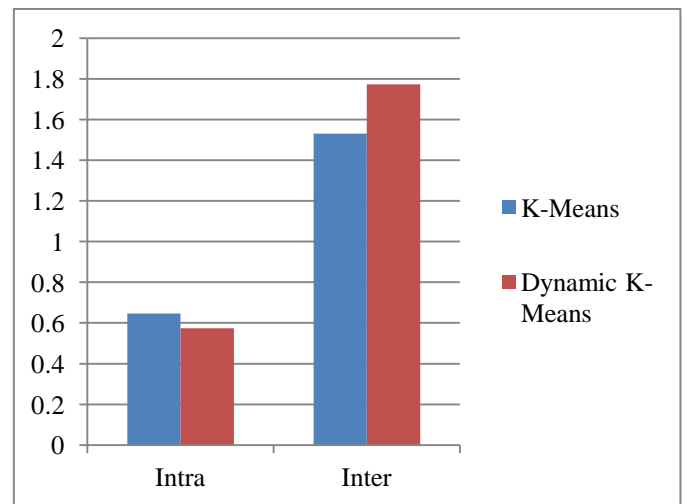


Sumber: Hasil Penelitian (2015)
Gambar 5. Aplikasi Segmentasi

Dari hasil percobaan yang dilakukan, algoritma *Dynamic K-means* dapat menghasilkan kualitas *cluster* yang lebih baik dibandingkan dengan *K-means*. Pada Tabel 3 menunjukkan bahwa algoritma *Dynamic K-means* menghasilkan jarak *inter* sebesar 1,7731 dan jarak *intra* sebesar 0,5737. Pada algoritma *K-means* jarak *inter* sebesar 1,5310 dan *intra* sebesar 0,6456. Jarak *inter* yang lebih besar dan *intra* yang lebih kecil menunjukkan bahwa algoritma *Dynamic K-means* dapat menghasilkan kualitas *cluster* yang lebih baik. Grafik hasil percobaan dapat dilihat pada Gambar 6.

Tabel 3. Hasil Percobaan

Algoritma	K	Intra	Inter
K-Means	3	0.6456	1.5310
Dynamic K-Means	5	0.5737	1.7731



Sumber: Hasil Penelitian (2015)

Gambar 6. Grafik Hasil Percobaan

Hasil evaluasi *cluster* menunjukkan bahwa algoritma *Dynamic K-means* memperoleh nilai *DBI* lebih kecil dibandingkan dengan algoritma *K-means* tradisional dengan nilai *DBI* sebesar 0.323558. Dengan nilai *DBI* yang lebih kecil menunjukkan bahwa skema *cluster* lebih optimal.

Tabel 4. Nilai Pengujian DBI

Algoritma	K	DBI
K-Means	3	0,421685
Dynamic K-Means	5	0,323558

Sumber: Hasil Penelitian (2015)

V. KESIMPULAN

Dari penelitian yang dilakukan, *Dynamic K-means*, terbukti dapat meningkatkan akurasi model yang terbentuk. Pengukuran *validity cluster* dengan menggunakan *Davies-Bouldin Index* (DBI), membuktikan bahwa *Dynamic K-means* menghasilkan kualitas *cluster* yang lebih optimal yang ditunjukkan dengan nilai DBI yang lebih kecil, dibandingkan dengan *K-means*. Nilai DBI yang lebih kecil mendekati 0 menunjukkan skema *cluster* yang optimal.

Berdasarkan hasil segmentasi pelanggan yang terbentuk, maka terdapat 5 kelompok pelanggan dengan karakteristik sebagai berikut:

Segmen 1 dengan jumlah pelanggan sebanyak 63 dengan urutan pembelian produk *grocery*, *milk*, *detergent*, *frozen*, *delicassen*, *fresh*.

Segmen 2 dengan jumlah pelanggan sebanyak 130 dengan urutan pembelian produk *grocery*, *milk*, *fresh*, *detergent*, *frozen*, *delicassen*.

Segmen 3 dengan jumlah pelanggan sebanyak 42 dengan urutan pembelian produk *fresh*, *milk*, *glocery*, *frozen*, *delicassen*, *detergent*.

Segmen 4 dengan jumlah pelanggan sebanyak 178 dengan urutan pembelian produk *fresh*, *glocery*, *milk*, *frozen*, *detergent*, *delicassen*.

Segmen 5 dengan jumlah pelanggan sebanyak 24 dengan urutan pembelian produk *fresh*, *glocery*, *milk*, *frozen*, *delicassen*, *detergent*.

REFERENSI

- [1] Adrian, R. Prepaid Telecom Customers Segmentation Using The K-Mean, *The Annals of The University of Oradea Economic Sciences*, 1112–1118. 2012
- [2] Aggarwal, N., & Aggarwal, K. Comparative Analysis of k-means and Enhanced K-means clustering algorithm for data mining, *International Journal of Scientific & Eninnering Research*, 3(3). 2012.
- [3] Chen, Y., Zhang, G., Hu, D., & Fu, C. Customer segmentation based on survival character. *Journal of Intelligent Manufacturing, IEEE* 18(4), 513–517. 2007
- [4] Deelers, S., & Auwatanamongkol, S. Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance, *World Academy of Science, Engineering and Technology*, 26(December), 323–328. 2007.
- [5] Lin, B., & Jones, C. Customer Segmentation Using K-Means Clustering and Decision Tree: A Research Review, *SouthWest Decision Sciences*. 2010
- [6] Maulik, U., & Bandyopadhyay, S. Performance Evaluation of Some Clustering Algorithms and Validity Indices, *IEEE Transaction On Pattern Analysis And Machine Intelligence*, 24(12), 1650–1654. 2002.
- [7] Myatt, G. J. *Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining*. Hoboken: John Willey & Sons. 2007.
- [8] Varcellis, Carlo. *Business Intelligence: Data Mining and Optimization for Decision Making*. Southrn Gate, Chichester, West Sussex: John Willey & Sons, Ltd. 2009.

- [9] Witten, I. H., Frank, E., & Hall, M. A. *Data Mining: Practical Machine Learning and Tool*. Burlington: Morgan Kaufmann Publisher. 2011.
- [10] Wu, Xindong & Kumar, Vipin. *The Top Ten Algorithms in Data Mining*. London: CRC Press. 2009.
- [11] Yi, B., Qiao, H., Yang, F., & Xu, C. An Improved Initialization Center Algorithm for K-Means Clustering, *International Conference on Computational Intelligence and Software Engineering, IEEE* (1), 1–4. 2010.
- [12] Zhang, C., & Fang, Z. An Improved K-means Clustering Algorithm Traditional K-mean Algorithm, *Journal of Information & Computational Science*, 1, 193–199. 2013.



Widiarina, M.Kom. Tahun 2008 lulus dari Program Strata Satu (S1) Program Studi Sistem Informasi STMIK Nusa Mandiri Jakarta. Tahun 2014 lulus dari Program Magister (S-2) Program Studi Manajemen Sistem Informasi STMIK Nusa Mandiri Jakarta. Saat ini bekerja sebagai tenaga pengajar di AMIK BSI Bekasi.