Peningkatan Hasil Cluster Menggunakan Algoritma Dynamic K-means dan K-means Binary Search Centroid

Gumilar Akbari, Yusrila Yeka Kerlooza

Magister Sistem Informasi, Fakultas Pasca Sarjana
Universitas Komputer Indonesia
Jalan Dipati Ukur no. 112 - 116, Bandung, Jawa Barat, Indonesia
✉ : gumilarakbarimail@gmail.com; kerlooza@unikom.ac.id

Abstrak — **Pada studi kasus segmentasi pelanggan, data yang digunakan untuk segmentasi memiliki atribut *data* berdasarkan nilai *Recency, Frequency,* dan *Monetery* dan memiliki jumlah data 500, untuk membentuk segmentasi pelanggan dapat digunakan teknik *clustering. Clustering* adalah proses untuk mengelompokkan datum ke dalam sejumlah *cluster* (kelompok data). Salah satu teknik *Clustering* adalah teknik *clustering partisi,* algoritma *clustering* yang digunakan pada penelitian ini yaitu algoritma Dynamic K-means (DK) dan K-means Binary Search Centroid (KBSC). Pada algoritma Dynamic K-means memiliki kemampuan untuk mencari jumlah *Cluster,* namun memiliki kekurangan dalam penentuan titik *centroid* (pusat cluster), sedangkan algoritma KBSC memiliki kemampuan untuk menentukan titik *centroid Cluster,* namun memiliki kekurangan dalam mencari jumlah *Cluster*. Pada penelitian ini menggabungkan kedua algoritma antara algoritma DK dan KBSC dan akan diujikan pada data model buatan yang bertujuan untuk melihat karakteristik dari algoritma, dan diujikan pada data studi studi kasus yang bertujuan untuk mengetahui kemampuan algoritma dalam menyelasaikan kasus segmentasi pelanggan. Berdasarkan pengukuran Devies Bouldin Index (DBI) algoritma gabungan DK-KBSC menghasilkan nilai DBI lebih baik dibandingkan algoritma lainnya.saat diimplementasikan pada data kasus segmentasi pelanggan.**

Kata Kunci — Segmentasi Pelanggan, *Clustering*, Dynamic K-Means, K-means, Binary Search Centroid.

I. PENDAHULUAN

Data Mining merupakan proses untuk mendapatkan informasi dari basis data untuk pengekstrakan informasi baru yang diambil dari data dengan jumlah yang besar untuk keperluan pengambilan keputusan [*1*]. Ada berbagai macam karakteristik data dilihat dari jumlah data dan jumlah atribut data, pada studi kasus penelitian ini akan melakukan segmentasi pelanggan distributor farmasi. Data yang digunakan memiliki karakteristik data dengan 3 atribut yang dilihat dari nilai *Recency* (jumlah hari yang dihitung dari tanggal terakhir transaksi ke tanggal acuan), *Frequency* (jumlah transaksi yang telah dilakukan), dan *Monetary* (total biaya yang telah dikeluarkan untuk transaksi) dengan jumlah data sebanyak 500 pelanggan. Tipe data memiliki nilai angka, yang diambil dari riwayat transaksi dalam periode satu tahun dari Januari sampai Desember di tahun 2016. Segmentasi akan dilakukan pada satu tahun terakhir dari riwayat transaksi ke tanggal acuan analisis.

Dalam kasus segmentasi maka terdapat cara penyelasaian yaitu dengan teknik Clustering. Clustering merupakan proses untuk mengelompokan suatu data dengan cara membandingkan kemiripan antar data, sehingga setiap data masuk ke dalam Cluster (grup) [*2*]. Clustering termasuk kedalam unsupervised learning (pembelajaran tidak terbimbing), karena ketika pengelompokan data didasarkan atas kemiripan dan ketidakmiripan antar datanya. Analisis cluster telah banyak digunakan termasuk riset pasar, pengenalan pola, analisis data, dan pengolahan citra. Dalam bisnis, clustering dapat membantu pemasar untuk menemukan kepentingan pelanggan mereka berdasarkan pada pola pembelian dan ciri kelompok pelanggan [*3*], oleh karena itu dalam kasus segmentasi pelanggan, maka diperlukan algoritma clustering yang cocok berdasarkan karakteristik dari datanya.

Untuk mengetahui algoritma clustering apa yang cocok digunakan pada segmentasi pelanggan, maka perlu mengetahui karakteristik dari beberapa algoritma clustering. Pada teknik partisi clustering terdapat algoritma K-means, algoritma K-means merupakan suatu algoritma clustering yang mempartisi dataset kedalam beberapa k cluster, algoritma K-means cukup mudah untuk diimplementasi dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan [*4*]. Proses segmentasi pelanggan terbanyak menggunakan algoritma K-means clustering, dan hasil menyatakan bahwa segmentasi sukses dan efektif [*5*]. Algoritma K-means digunakan untuk kasus segmentasi pelanggan karena penentuan jumlah cluster dapat disesuaikan dengan kebutuhan sehingga jumlah cluster dapat diasumsikan diawal proses cluster. namun algoritma K-means mempunyai kelemahan yaitu harus dapat menduga jumlah cluster diawal [*6*], karena asumsi jumlah cluster yang ditentukan diawal belum tentu menghasilkan cluster baik untuk hasil segmentasi. Kemudian algoritma K-means lainnya yaitu dalam penentuan titik centroid awal yang masih dipilih secara acak [*7*] yang akan berpengaruh pada kondisi local minima (kondisi hasil cluster belum tentu yang terbaik). Penentuan titik centroid akan berpengaruh pada iterasi proses untuk menghasilkan cluster convergence yang artinya hasil cluster memiliki anggota cluster yang tidak berubah [*8*].

Pada saat kondisi dimana data segmetasi pelanggan tidak ingin ditentukan sesuai kebutuhan, namun ingin menghasilkan segmentasi dari sebaran data, maka untuk penentuan asumsi jumlah cluster tidak diduga diawal, namun harus dicari asumsi jumlah cluster yang akan terbentuk. Untuk menangani kondisi tersebut, terdapat pengembangan dari algoritma K-means, yaitu algoritma Dinamyc K-means (DK) yang memiliki proses untuk mencari jumlah cluster tanpa harus menduga asumsi jumlah cluster [*9*], Pada penelitian Ahamed Shafeeq dan

Hareesha hasil cluster memiliki kualitas cluster yang baik berdasarkan nilai intra dan inter cluster, dibandingkan dengan algoritma K-means namun memiliki beberapa kekurangan yang sama dengan algoritma K-means yaitu masalah pada penentuan titik centroid pada proses cluster yang masih dipilih secara acak.

Berdasarkan penelitian dari Yugar Kumar dan G.Sahoo untuk masalah penentuan titik centroid dilakukan dengan cara mengembangkan K-means menjadi K-means Binary Search Centroid (KBSC) yang memiliki proses dalam menenentuan titik centroid menggunakan pendekatan teknik *Binary Search* [*8*]. Pada hasil penelitian oleh Yugar Kumar dan G Sahoo, algoritma KBSC diuji pada data Iris,Wine, dan Diabetes dengan masing-masing jumlah atribut (4, 13 dan 8), berdasarkan hasil percobaan Algoritma KBSC memiliki nilai *intra* dan *inter cluster* yang lebih baik dari algoritma K-mean, dan juga jika dilihat berdasarkan akurasi *cluster* algoritma KBSC baik digunakan pada data dengan jumlah atribut = 4 dan jumlah data 150 yaitu pada hasil data Iris jika dibandingkan dengan percobaan data Wine dan Diabetes. Algoritma KBSC memiliki keterbatasan dalam menentukan asumsi jumlah *cluster* yang akan dibentuk.

II. TINJAUAN PUSTAKA

Pada bagian ini akan menjelaskan mengenai penggabungan dari dua metode yang ada pada clustering, algoritma yang digabungkan yaitu Dinamiy K-means (DK) dan K-means Binary Search Centroid (KBSC). Berikut ini penjelasan masing-masing metode:

A. *Algoritma DK*

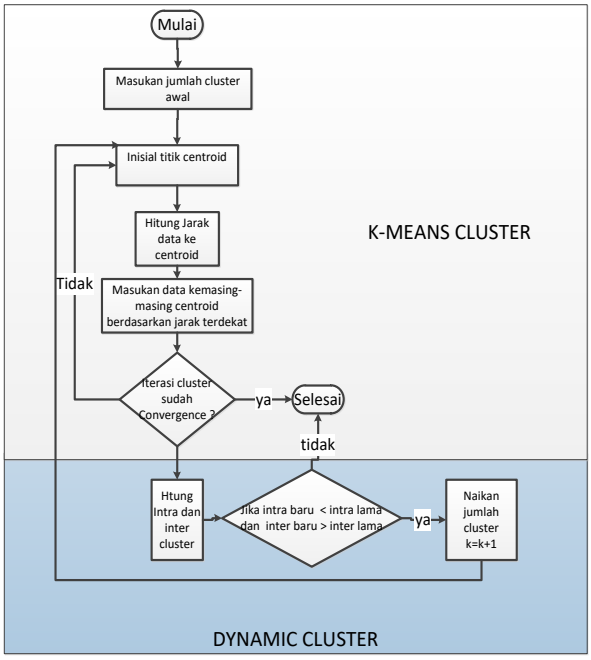
Dynamic K-means (DK) adalah algoritma yang dikembangkan dari algoritma K-means. Cara kerjanya sama dengan algoritma K-means, namun ketika anggota pada satu *cluster* sudah tidak berubah atau tidak berpindah ke *cluster* lain dilakukan perhitungan jarak *intra* dan *inter cluster*. Jika jarak *intra* semakin kecil dan jika jarak *intra* semakin besar, maka algoritma akan menghitung *cluster* baru dengan menambahkan jumlah *cluster* dengan satu atau *k=k+1* disetiap iterasi.

Penjelasan dari Gambar 1 sebagai berikut ini :

1. Masukan jumlah *cluster* (*k*).
2. Menentukan titik *centroid* data.
   Penentuan awal titik *centroid* data *cluster* ini bisa dilakukan dengan berbagai cara, namun yang paling sering dilakukan adalah dengan cara acak oleh proses algoritma.
3. Kelompokan semua data atau objek yang memiliki kemiripan dengan cara menghitung jarak ke titik *centroid* terdekat. Jarak ini diartikan sebagai kesamaan suatu objek data dengan titik *centroid* yang telah ditentukan. Kedekatan dua objek ditentukan berdasarkan perhitungan jarak objek tersebut. Untuk menghitung jarak semua data ke setiap titik *centroid cluster* dapat menggunakan teori jarak *Euclidean Distance* yang dirumuskan sebagai berikut :

$$D(i,j) = \sqrt{(x_{1i} - x_{1j})^2 + (x - x_{2j})^2 + \cdots + (x - x_{kj})^2} \quad (1)$$

Keterangan:
D(i, j) : Jarak data ke i ke *centroid Cluster* j
x_{kt} : Data ke i pada variabel data ke k
x_{kj} : Titik *centroid* ke j pada variabel ke k



Gambar 1. Algoritma DK

4. Hitung kembali titik *centroid* baru berdasarkan keanggotaan yang terbentuk. Titik *centroid* baru adalah rata-rata dari semua data atau objek dalam *cluster* tertentu.
5. Jika anggota *cluster covergence* tidak berubah maka lakukan pada proses hitung intra *cluster* dan juga inter *cluster*.
6. Jika *intra* baru < *intra* lama dan *inter* baru > *inter* lama, maka jumlah *cluster* akan ditambahkan dengan *k=k+1* dan kembali pada proses kedua menentukan titik *centroid* data dan mengalokasikan data. Jika tidak maka proses *cluster* akan berhenti pada k=k.

B. *Algoritma KBSC*

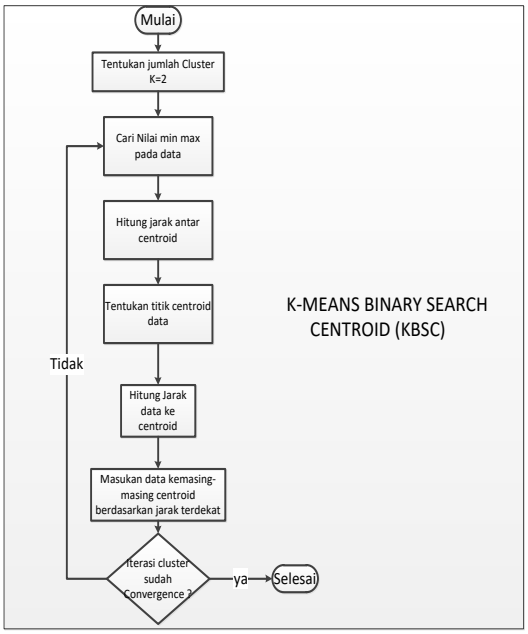
Algoritma KBSC merupakan pengembangan dari algoritma *K-mean*s dengan cara penggabungan antara algoritma K-means dengan algoritma tambahan pada area proses pembentukan titik *centroid* data menggunakan pendekatan *Binary Search* [8].

Penjelasan Gambar 2 adalah sebagai berikut ini :

1. Tentukan jumlah *cluster* (k) nilai k=1, 2, 3… m.
2. Hitung nilai maximum dan minimum pada data untuk masing-masing atribut data.
3. Hitung range antara titik *centroid*, perhitungan range antar titik *centroid* pada suatu data dengan cara berikut ini :

$$M = \frac{max(a_i) - min(a_i)}{k} \qquad (2)$$

Berdasarkan rumus 2 digunakan untuk menghitung suatu nilai pada variabel M yang merupakan jarak spesifik antar titik *centroid data* sebelum memberikan hasil titik *centroid data*. $min(a_i)$ nilai minimal dari masing-masing atribut data, $max(a_i)$ merupakan nilai maksimum dari masing-masing atribut data, *k* adalah jumlah *cluster* yang akan dibentuk.



Gambar 2. Algoritma KBSC

4. Tentukan titik *centroid* data, Untuk menghasilkan titik *centroid* menggunakan rumus sebagai berikut :

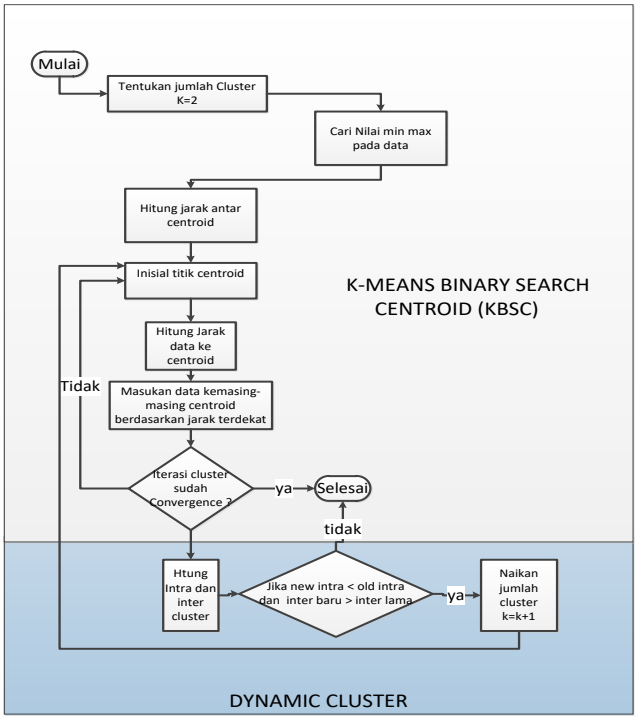
$$C_k = \min(a_i) + (k-1)M \qquad (3)$$

Berdasarakan rumus 3 , C_k merupakan *centroid* untuk *cluster k*, $\min(a_i)$ nilai minimum data, dan M range antara titik *centroid* data.

5. Lakukan pengelompokan data dengan cara hitung jarak data terhadap titik *centroid* yang terbentuk Ck. Perhitungan jarak menggunakan *Euclidiance distance* pada rumus:
6. Hitung kembali titik *centroid* berdasarkan *cluster* yang terbentuk, Kemudian lakukan pengelompokan ulang sampai menemukan *cluster* tidak berubah kembali *(convergence).*

C. Algoritma Gabungan DK-KBSC

Pada penelitian ini akan menggabungkan algoritma DK dan KBSC dengan kelebihan masing-masing algoritma untuk mencari jumlah *cluster* terbaik, dan menentukan titik centroid awal pada setiap pembentukan jumlah *cluster*. Berikut ini gambar mengenai penggabungan kedua algortima:



Gambar 3. Alur Penggabungan Kedua Algoritma

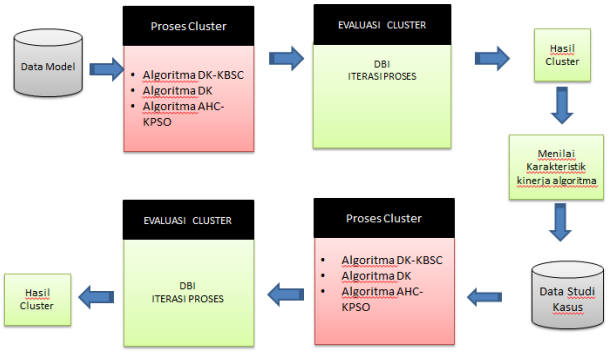
Pada Gambar 3 menjelaskan alur penggabungan dari kedua algoritma. *Flowchart.* Titik *centroid* ini akan menggunakan teknik pencarian data dengan pendekatan *Binary Search* (Pencarian bagi dua terhadap data). Proses tersebut akan membagi area data berdasarkan jumlah *cluster* yang dimasukan. Berikut ini penjelasan pada alur *flowchart*:

1. Masukan jumlah *cluster* sebanyak *k*
2. Setelah itu proses dilanjutkan ke proses penentuan titik *centroid* menggunakan pendekatan *binary search centroid*
3. Kemudian lakukan pengelompokan dengan cara menghitung jarak *minimum* antar data dengan *titik centroid* menggunakan *Euclidiance Distance* sehingga tidak ada lagi titik data yang berpindah *cluster*. Di rumus 2.1 menjelaskan perhitungan jarak *Euclidiance Distance*
4. Setelah *cluster* tidak berubah, tahap selanjutnya masuk pada tahap proses *Dynamic K-means* yang mana prosesnya menghitung nilai *inter* dan nilai *intra*. Dan akan ditambahkan jumlah k=k+1 jika nilai *intra* baru < *intra* lama, dan *inter* baru > dari *inter* lama. Dan akan kembali ke proses penentuan titik *centroid* kembali berdasarkan *cluster* yang terbentuk sebelumnya.

III. METODOLOGI PENELITIAN

Algoritma usulan akan dilakukan pada data model dan data kasus. Hasil pengujian digunakan suatu penilaian secara *kuantitatif*, yang mana penilaian tersebut merupakan hasil perbandingan antara gabungan algoritma yang diajukan dengan algoritma lain, yang memiliki tujuan untuk dapat mengetahui kemampuan dan karakteristik algoritma dalam menyelesaikan kasus *clustering* pada bentuk data tertentu.

Berikut ini alur pengujian terhadap penggabungan algoritma pada data yang digunakan :



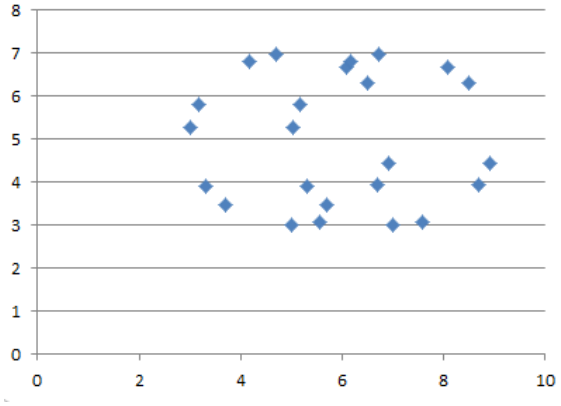
Gambar 4. Tahapan Pengujian dan Proses Perbandingan Kinerja Algoritma

Penjelasan berdasarkan Gambar 4 adalah setelah data diproses oleh algoritma yang diusulkan, maka akan muncul suatu nilai evaluasi *clustering* yang digambarkan menggunakan grafik perbandingan. Sehingga dapat mengetahui kelebihan maupun kekurangan algoritma terhadap data yang diuji.

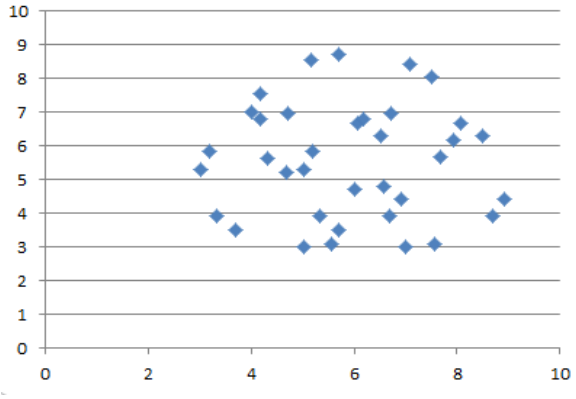
Data model uji merupakan data yang dibuat dengan bentuk yang memiliki sebaran dengan nilai kemiripan intra dan inter yang telah diatur, dan bentuk datanya dibuat dalam beberapa sebaran bentuk *cluster*. Sebaran bentuk *cluster* tersebut dilihat berdasarkan jarak intra dan inter. Pada penelitian ini, sebaran untuk data model akan dibuat sebagai berikut:

A. *Sebaran Data Model Bentuk Lingkaran*

Sebaran data bentuk lingkaran memiliki variasi nilai *intra* = 2 dan *inter* = *2* , dan jumlah variasi bentuk *cluster* yang berbeda mulai dari 2 *cluster* sampai 3 *cluster* yang akan ditetapkan. Tujuan dari data model buatan ini yaitu untuk mengetahui karakteristik kinerja dari algoritma gabungan DK-KBSC berdasarkan nilai DBI yang akan dihasilkan. Berikut ini data model yang akan diuji pada penelitian ini:



Gambar 5. Bentuk Sebaran Data 2 Cluster (2 Lingkaran)



Gambar 6. Data Model Sebaran Bentuk Lingkaran

Berikut ini penjelasan dari data sebaran lingkaran yang ditentukan berdasarkan jarak intra dan inter cluster dan dari variasi jumlah data dan jumlah atribut:

Tabel 1. Data Model Buatan Sebaran Lingkaran

Sebaran *Cluster*	**Jarak Intra / inter *cluster***	**Jumlah Variabel**	**Jumlah Data**
2 lingkaran	2/2	<=13	<=768
3 lingkaran	2/2	<13	<=1152

Pada penelitian ini hanya akan berfokus pada data yang jarak intra dan inter *cluster*nya berdekatan, karena jika intra dan inter nya berdekatan belum dapat diketahui jumlah *cluster* yang dapat terbentuk seharusnya.

B. Data Model dari Machine Learning Dataset

Data yang akan digunakan dalam penelitian adalah *iris, diabetes dan wine*. Untuk proses *clustering* menggunakan penggabungan algoritma Dyanamic K-means dan KBSC, pada ketiga dataset yang dicoba ini memiliki kesamaan pada tipe datanya adalah *numeric*. Data testing diperoleh dari UCI *repository of machine learning databases*, yang disediakan oleh Nationan Science Foundation yang berisi kumpulan *dataset* untuk keperluan penelitian di bidang *data mining.*

Data testing ini dipilih karena memiliki tipe data yang sesuai untuk proses *Clustering*, dan memiliki keragaman pada atribut dan jumah datanya yang akan dijadikan acuan pada proses *cluster*. Pertimbangan dalam memilih data testing, karena telah banyak digunkan dan diuji oleh beberapa penelitian sebelumnya mengenai kasus *clustering*. Berikut ini tabel mengenai karakteristik dari masing-masing dataset yang akan digunakan:

Tabel 2 Data Model UCI

Data Set	**Variabel (Atribut Data)**	**Jumlah Data**	**TIpe Data**
Iris	4	<= 768	*Numeric*
Diabetes	8	<= 1152	
Wine	13	<= 768	

C. *Data Model dari Machine Learning Dataset*

Data uji studi kasus memiliki karakteristik data dengan jumlah atribut yaitu 3 atribut, dimana ketiga atribut tersebut masing-masing memiliki nilai numeric.Jumlah data yang akan diuji yaitu 500 pelanggan dengan nilai atibut masing-masing yaitu *Recency*, *Frequency* dan *Monetary*.

Pada kasus ini ingin mengetahui segmentasi pelanggan dengan mengacu pada nilai RFM yang didapat dari data historis transaksi pelanggan, sehingga pihak perusaahaan khususnya manager pemasaran dapat membuat layanan pemasaran yang berbeda agar sesuai dengan kebutuhan dan keinginan setiap segmen pelanggan.

Berdasarkan dari data historis transaksi, akan diambil transaksi selama satu tahun dimulai tanggal 1 januari 2016 sampai 1 Januari 2017, kemudian diambil nilai RFM pelanggan tersebut. Jumlah sampel data pelanggan yang akan diproses *cluster* sebanyak 500 pelanggan. Di Tabel 19 merupakan data yang akan di proses *cluster* untuk menghasilkan segmentasi pelanggan.

IV. HASIL DAN PEMBAHASAN

Pada pengujian ini diuji untuk sebaran data sebagai 2 c*luster* dan 3 *cluster* dengan tingkatan jumlah data dan jumlah variable. Tujuan dari pengujian ini untuk mengetahui kinerja algoritma dalam melihat karakteristik dengan jumlah data dan variable yang berbeda berdasrkan hasil DBI *cluster* yang dihasilkan.

Pengujian data dilakukan dengan membandingkan algoritma gabungan DK-KBSC dengan algoritma DK bertujuan untuk mengetahui hasil DBI cluster dari centroid acak pada algoritma DK, dengan hasil DBI centroid Binary Search pada algoritma DK-KBSC, hal ini dilakukan berdasarkan pada penelitian algoritma KBSC yang membandingkannya dengan algoritma sebelumnya. Pada penelitian ini juga akan membandingkan hasil pengujian antara algoritma DK-KBSC dengan AHC-KPSO bertujuan untuk mengetahui nilai DBI dengan teknik inisialisasi *centroid* yang berbeda.

1) Pengujian DBI Sebaran Data 2 Lingkaran Dekat

Hasil DBI pada data lingkaran 2 c*luster* akan diuji dengan tingkatan jumlah data yang dimulai dengan 24 data sampai dengan 768, dan juga tingkatan jumlah variable sebagai berikut ini:

a. Data Sebaran Dua Lingkaran Dekat 2 Variabel (x,y)

Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI:

Tabel 3. Hasil Uji DBI *Cluster* Pada Sebaran Data Dua Lingkaran Dengan 2 Variabel (x,y)

Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	**DK**	**AHC – KPSO**
Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	0.737428	0.645432245
Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	0.748589	0.693062529
Berikut ini kinerja masing-masing algoritma untuk data	Berikut ini kinerja masing-masing algoritma untuk data	0.774989	0.759292473

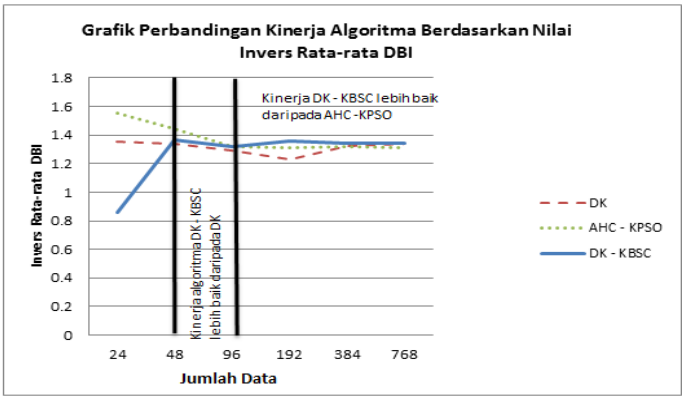
lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :		
Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	0.814454	0.762506091
Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	0.7544725	0.757192413
Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	Berikut ini kinerja masing-masing algoritma untuk data lingkaran 2 *cluster* dengan 2 variabel dalam visualisasi grafik berdasarkan invers dari nilai rata-rata DBI :	0.746964	0.763067088

Pada setiap jumlah data, dilakukan beberapa kali percobaan sehinggan pengujian DBI cluster memiliki standar deviasi sebagai berikut:

Tabel 4 Standar Deviasi DBI *Cluster* pada Sebaran Data Dua Lingkaran Dengan 2 Variabel (x,y)

Jumlah Data	Algoritma		
	DK - KBSC	DK	AHC – KPSO
24	0	0.029979971	0
48	0	0.037415629	0.001276
96	0	0.04264984	0.00251
192	0	0.125179835	0.00426
384	0	0.012587608	0.010413
768	0	0.001422139	0.01459

Berikut ini kinerja masing-masing algoritma dalam visualisasi grafik berdasarkan invers rata-rata DBI:



Gambar 7. Hasi Grafik Perbandingan Kinerja Algoritma Pada Sebaran Data Dua Lingkaran Dekat (2 Variabel)

Penjelasan dari Gambar 7 bahwa algoritma DK–KBSC jika dibandingkan dengan algoritma AHC-KPSO memiliki kinerja yang baik pada data dengan jumlah data >=96, sedangkan jika dibandingkan dengan algoritma DK, kinerja algoritma DK–KBSC akan baik pada data dengan jumlah >=48.

b. Data Sebaran Dua Lingkaran Dekat 3 Variabel (x,y,z)

Hasil invers rata-rata DBI dan jumlah iterasi dengan jumlah variable = 3 akan dijelaskan di dan memiliki nilai deviasi sebagai berikut ini:

Tabel 5. Invers Rata-rata DBI Yang Dihasilkan Pada Sebaran Data Dua Lingkaran Dengan 3 Variabel (x,y,z)

Jumlah Data	Algoritma		
	DK –KBSC	DK	AHC – KPSO
24	1.01619	1.063918	0.920631
48	1.03744	1.041825	0.907822
96	1.03968	1.04111	1.036055
192	1.0334	1.047412	1.099998

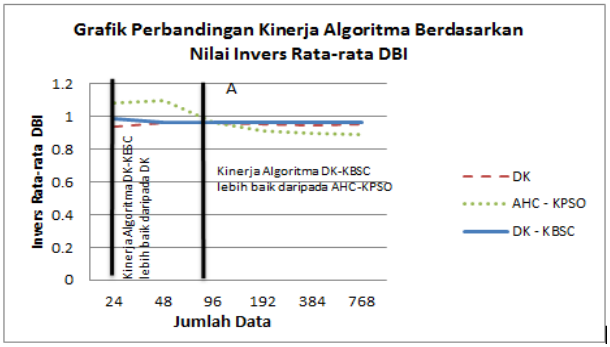
384	1.03721	1.053776	1.110991
768	1.03258	1.044944	1.122307

Pada setiap jumlah data, dilakukan beberapa kali percobaan sehinggan pengujian DBI cluster memiliki standar deviasi sebagai berikut:

Tabel 6. Standar Deviasi Rata-rata DBI Pada Sebaran Data Dua Lingkaran Dengan 3 Variabel (x,y,z)

Jumlah Data	Algoritma		
	Standar Deviasi	Standar Deviasi	Standar Deviasi
24	0	0.229449	0
48	0	0.004752	0.421637
96	0	0.001038	0.816497
192	0	0	1.286684
384	0	0	1.264911
768	0	0	2.973961

Berikut ini kinerja masing-masng algoritma dalam visualisasi grafik berdasarkan invers rata-rata DBI:



Gambar 8. Grafik Kinerja Algoritma Pada Sebaran Data Dua Lingkaran Dengan 3 Variabel (x,y,z)

Penjelasan dari Gambar 8 algoritma usulan DK– KBSC jika dibandingkan dengan algoritma AHC-KPSO memiliki kinerja yang baik pada data dengan jumlah data >=96, sedangkan jika dibandingkan dengan algoritma DK, kinerja tersebut akan baik pada data dengan jumlah data >=48, hasil ini dilihat berdasarkan pengukuran nilai DBI *cluster*.

Untuk mengetahui presisi algoritma DK – KBSC maka diukur menggunakan standar deviasi DBI yang ada di Tabel 4. Berdasarkan Tabel 4 pada saat pengukuran nilai DBI dari algoritma usulan DK – KBSC, memiliki presisi nilai DBI yang lebih baik dari pada AHC - KPSO dan algoritma DK dengan nilai deviasi = 0.

2) Pengujian Data UCI

Hasil uji DBI cluster masing-masing algoritma dapat dilihat sebagai berikut ini:

a. Data Iris

Tabel 7. Invers rata-rata DBI Uji Data Iris

Jumlah Data	Algoritma		
	DK-KBSC	DK	AHC-KPSO
150	0.40483	0.6641	0.40483
192	0.44203	0.7278	0.44203
288	0.49152	0.8538	0.49152
384	0.53896	0.8390	0.53896
576	0.53896	0.9575	0.53896
768	0.5491	0.9294	0.5491

Pada setiap jumlah data, dilakukan beberapa kali percobaan sehinggan pengujian DBI cluster memiliki standar deviasi sebagai berikut:

Tabel 8. Standar Deviasi DBI Uji Data Iris

Jumlah Data	Algoritma		
	DK-KBSC	DK	AHC-KPSO
150	0	4.5092498	0.7071068
192	0	2.0615528	1.1595018
288	0	3.6855574	1.2649111
384	0	6.25	1.1737878
576	0	2.0615528	1.9888579
768	0	5.6199051	2.0976177

b. Data Wine

Tabel 9. Invers rata-rata DBI Uji Data Wine

Jumlah Data	DK –KBSC		
	DK-KBSC	DK	AHC-KPSO
178	0.4817	0.4814	0.4817
192	0.4919	0.4919	0.4919
288	0.5202	0.5202	0.5455
384	0.5257	0.5258	0.5555
576	0.5202	0.5202	0.5192
768	0.5180	0.5191	0.5491

Pada setiap jumlah data, dilakukan beberapa kali percobaan sehinggan pengujian DBI cluster memiliki standar deviasi sebagai berikut:

Tabel 10. Standar Deviasi DBI Uji Data Wine

Jumlah Data	DK –KBSC		
	DK-KBSC	DK	AHC-KPSO
178	0.4817	0.4814	0.4817
192	0.49193	0.4919	0.4919
288	0.52026	0.5202	0.5455
384	0.52579	0.5258	0.5555
576	0.52026	0.5202	0.5192
768	0.51803	0.5191	0.5491

c. Data Wine

Tabel 11. Invers rata-rata DBI Uji Data Diabetes

Jumlah Data	Algoritma		
	DK-KBSC	DK	AHC-KPSO
24	0.3408	0.3408	1.6202
48	0.4844	0.4844	1.5655
96	0.5885	0.6648	1.6817
192	0.6205	0.6253	1.5991
384	0.6426	0.6503	1.5376
768	0.6290	0.6506	1.4017

Pada setiap jumlah data, dilakukan beberapa kali percobaan sehinggan pengujian DBI cluster memiliki standar deviasi sebagai berikut:

Tabel 12 Standar Deviasi DBI Uji Data Diabetes

Jumlah Data	Algoritma		
	DK-KBSC	DK	AHC-KPSO
24	0	0	0
48	0	0	0
96	0	0	0
192	0	0	0
384	0	1.17027	0
768	0	0.0194194	0

Berdasarkan Tabel 7 menjelaskan bahwa berdasarkan nilai invers *DBI*, algoritma usulan gabungan antaran DK–KBSC jika diimplementasikan pada data set, kinerja algoritma tersebut memiliki kesamaan dibandingkan dengan AHC – KPSO *dan* DK.

Berdasarkan Tabel 9 menjelaskan bahwa berdasarkan nilai invers *DBI*, algoritma usulan gabungan antaran DK–KBSC jika diimplementasikan pada data set Wine, kinerja algoritma tersebut akan baik pada data dengan jumlah >=192 data dibandingkan dengan algoritma AHC–KPSO, dan memiliki kinerja yang sama pada setiap jumlah data jika dibandingkan dengan DK namun iterasi yang dihasilkan oleh algoritma usulan DK–KBSC lebih kecil pada jumlah data tertentu.

Berdasarkan Tabel 11 menjelaskan bahwa berdasarkan nilai invers *DBI*, algoritma usulan gabungan antaran DK–KBSC jika diimplementasikan pada data set Wine, kinerja algoritma tersebut akan baik pada seluruh jumlah data uji data jika dibandingkan dengan algoritma AHC-KPSO, dan memiliki kinerja yang baik pada data dengan jumlah 48 sampai 768 jika dibandingan dengan kinerja algoritma DK.

Berdasarkan Standar Deviasi *DBI cluster* masing-masing algoritma, algoritma gabungan DK-KBSC memiliki tingkat presisi yang palin baik karena memiliki standar deviasi =0 pada setiap percobaannya untuk masing-masing data Iris, Wine, dan Diabetes.

3) Pengujian Data Kasus

Berdasarkan hasil pengujian data kasus segmentasi menghasilkan hasil cluster sebagai berikut ini:

Tabel 13. Hasil Uji Data Kasus Segmentasi Pelanggan

Algoritma	*Jumlah Cluster*	DBI		Iterasi	
		DBI	St.Dev	Iterasi	St.Dev
DK	3	0.30909	0	12.57	1.511
AHC– KPSO	2	0.28259	0	8	0
DK-KBSC	4	0.27238	0	4	0

Hasil *Cluster* oleh algoritma gabungan DK-KBSC memiliki nilai DBI terbaik karena memiliki nilai DBI terkecil dibandingkan dengan algoritma DK, dan AHC-KPSO dan memiliki jumlah iterasi yang lebih kecil dibandingkan algoritma lainnya, untuk melihat presisi DBI, algoritma DK-KBSC lebih baik dibandingkan algoritma DK, dan memiliki kesamaan dengan algoritma AHC-KPSO.

V. KESIMPULAN

Berdasarkan beberapa percobaan yang dilakukan, kesimpulan dari penggabungan algoritma DK-KBSC sebagai berikut:

1. Berdasarkan pengujian data model, nilai DBI algoritma usulan DK- KBSC memiliki kinerja yang baik pada data yang memiliki jumlah data >=48, berdasarkan pengujian terhadap dengan sebaran bentuk 2 lingkaran dekat dengan jarak *intra cluster* = 2 dan *inter cluster*= 2.
2. Pada data model, kinerja algoritma usulan DK-KBSC memiliki hasil iterasi yang kecil dibandingkan dengan AHC–KPSO, hasil tersebut pada data dengan jumlah 768 dan 1152 pada data model, dan juga pada data set. Sedangkan pada kondisi tertentu iterasi dari algoritma DK terkadang lebih kecil dari algoritma usulan DK – KBSC. Tetapi hasil jumlah iterasi pada algoritma usulan memiliki kebaikan pada nilai deviasinya dibandingkan dengan DK.
3. Algoritma usulan DK-KBSC memiliki kelemahan pada jumlah variabel yang lebih dari 3 variabel, hasil tersebut berdasarkan nilai DBI yang cenderung memiliki kinerja yang sama atau memiliki kinerja dibawah dengan algoritma lain adalah AHC - KPSO.
4. Berdasarkan pengujian data kasus dengan jumlah atribut 3 dan jumlah data 500, algoritma DK-KBSC memiliki nilai DBI yang paling kecil dan dengan jumlah iterasi yang paling kecil, dan juga dengan presisi yang lebih baik dibandingkan algorima DK.

Dalam penelitian ini terdapat beberapa hal yang dapat di tinjau kembali untuk keperluan penelitian sebagai berikut:

1. Data yang diuji lebih beragam lagi dan bervariasi, sehingga dapat lebih mengatahui kinerja gabungan algoritma DK-KBSC.
2. Pada penggabungan algoritma ini tidak memperhatikan masalah *outlie r*(data yang jaraknya tidak rapat, namun masuk dalam satu *cluster*) pada *cluster*, oleh karena itu dapat digabungkan dengan algoritma lain dalam masalah outlier.
3. Dalam penelitian ini tidak mempertimbangan teknik pemrograman, diperlukan teknik pemrograman yang lebih

baik, sehingga proses bisa lebih baik dan dapat mengangani data dengan jumlah data yang lebih banyak.

DAFTAR PUSTAKA

[1] Eko Prasetyo, *Data Mining Konsep dan Aplikasi Menggunakan MATLAB*. Yogyakarta: ANDI, 2012.

[2] Budi Santosa, *Data Mining Teknik dan Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: GRAHA ILMU, 2007.

[3] Arpit Bansal , Mayur Sharma , and Shalini Goel , "Improved K-mean Clustering Algorithm for Prediction Analysis using Classification Technique in Data Mining," *International Journal of Computer Applications (0975 – 8887)*, vol. 157, p. 35, January 2017.

[4] Wu Xindong and Vivin Kumar, *The Top Ten Algorithms in*. London: CRC Press., 2009.

[5] Ye , Lou , Qiu-ru , Cai , and Etl. , "Telecom Customer Segmentation with," in *the 7th International Conference Computer Science & Education (ICCSE)*, Melbourne, VIC, 2012.

[6] Yudi Agusta, "K-Means – Penerapan, Permasalahan," *Jurnal Sistem dan Informatika*, vol. 3, p. 51, Februari 2007.

[7] Madhu Yedla, Srinivasa Rao Pathakota, and T M Srinivasa, "Enhancing K-Means Clustering Algorithm with Improved Initial Cluster," *International Journal of Computer Science and Information Technologies*, vol. 1, no. 2, pp. 121-125, 2010.

[8] Kumar,Yugar and G Sahoo , "A New Initialization Method to Originate Initial Cluster Centers for," *International Journal of Advanced Science and Technology*, p. 44, 2014.

[9] Ahmed Shafeeq and Hareesha K M, "Dynamic Clustering of Data with Modified K-Means Algorithm ," *ICICN*, vol. 27, pp. 221-225, 2012.