BAB 4
HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Pengujian Karakteristik Algoritma

Hasil pengujian karakteristik algoritma terdiri dari hasil pengujian data model dan data testing, kemudian dilakukan analisis perbandingan kinerja algoritma berdasarkan hasil pengujian data model dan data testing.

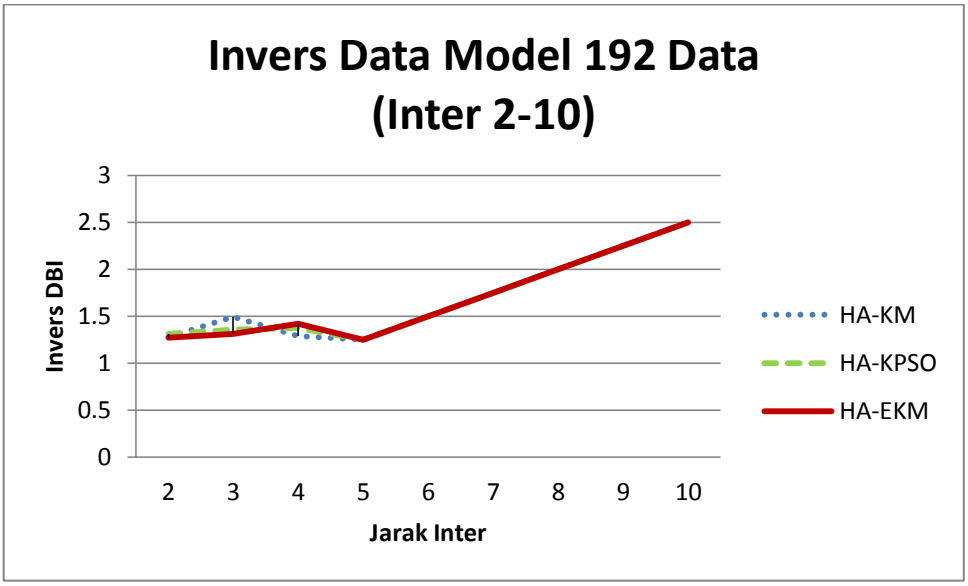
4.1.1 Hasil Pengujian Data Model

Pengujian data model diawali dengan mencari perilaku algoritma dalam membentuk hasil *clustering* untuk *dataset* dengan jarak inter yang dekat dan yang jauh. Tabel 4.1 menunjukkan salah satu hasil percobaan yang dilakukan pada *dataset* 2 atribut – 2 *cluster* yang berjumlah 192 data untuk jarak inter 1 sampai 10.

Tabel 4.1 Nilai DBI Data Model 192 Data (Inter 2-10)

Jarak Inter	Nilai DBI		
	HA-KM	HA-KPSO	HA-EKM
2	0.78066	0.76250	0.784793
3	0.670314	0.73623	0.7612
4	0.775588	0.72678	0.704198
5	0.799832	0.799832	0.799832
6	0.666527	0.666527	0.666527
7	0.571309	0.571309	0.571309
8	0.499895	0.499895	0.499895
9	0.444351	0.444351	0.444351
10	0.399916	0.399916	0.399916

Tabel 4.1 memperlihatkan hasil evaluasi *clustering* DBI algoritma HA-KM,HA-KPSO dan HA-EKM. Untuk mempermudah perbandingan nilai DBI, hasil tersebut divisualisasikan ke dalam grafik yang dapat dilihat di Gambar 4.1.



Gambar 4.1 Invers DBI Data Model 192 Data (Inter 2-10)

Pada Gambar 4.1 terlihat bahwa pada *dataset* dengan jarak inter di bawah 5, nilai DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM berbeda-beda, sementara pada jarak inter 5 hingga 10, ketiga algoritma memiliki nilai DBI yang sama. Hal ini menunjukkan bahwa pada saat jarak antar titik pusat *cluster* berdekatan (bentuk *cluster* masih acak) ketiga algoritma menghasilkan kinerja yang berbeda-beda. Sementara saat jarak antar titik pusat *cluster*-nya berjauhan, ketiga algoritma memiliki kinerja yang sama sebab bentuk *cluster* sudah terbentuk dengan baik sehingga jumlah *cluster* yang terdeteksi adalah benar yaitu 2 *cluster*. Berdasarkan hal tersebut maka untuk selanjutnya, pengujian difokuskan pada *dataset* dengan jarak inter dekat yaitu 2.

Pengujian data model ini dilakukan dengan menggunakan evaluasi hasil *clustering* DBI terhadap hasil *clustering* algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* model. Hasil pengujian yang ditampilkan merupakan hasil yang dapat menunjukkan atau mewakili karakteristik algoritma. Hasil ini diperoleh dari hasil pengujian *dataset* 2 *cluster* dengan jumlah atribut 2,3 dan 4. Hasil pengujian terhadap *dataset* dengan jumlah atribut lainnya dapat dilihat di lampiran.

Hasil *clustering* yang diperoleh dari algoritma HA-KM dan HA-KPSO bergantung pada penentuan pusat *cluster* awal yang dipilih secara acak, sehingga hasil evaluasi yang didapatkan nilainya tidak tetap. Sehingga dilakukan 10 kali percobaan untuk setiap algoritma lalu diambil rata-ratanya untuk digunakan dalam perbandingan kinerja. Selain itu, dicari juga standar deviasinya yang dapat menunjukkan bahwa semakin kecil standar deviasi maka semakin presisi algoritma tersebut.

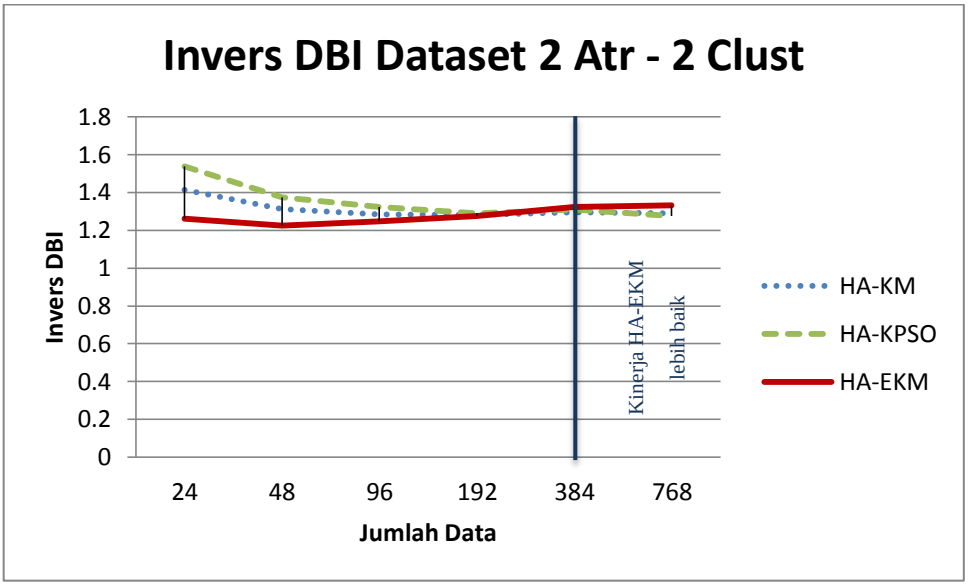
A. Hasil Pengujian *Dataset* 2 Atribut

Evaluasi hasil *clustering* DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* 2 atribut – 2 *cluster* dapat dilihat di Tabel 4.2 sementara grafik kinerjanya dapat dilihat di Gambar 4.2 dan Gambar 4.3 untuk standar deviasinya.

Tabel 4.2 Nilai DBI dan Standar Deviasi *Dataset* 2 Atribut – 2 *Cluster*

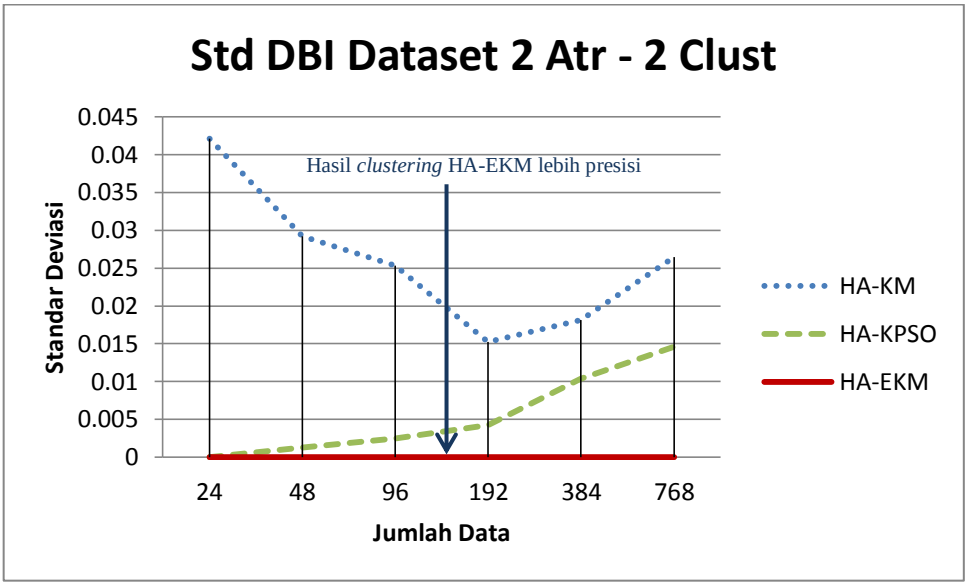
Jumlah data	DBI			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	0.707302	0.649928	0.792465	0.04207	0	0
48	0.761932	0.727653	0.816866	0.029221	0.001276	0
96	0.778495	0.755331	0.802697	0.025323	0.00251	0
192	0.78066	0.774897	0.784793	0.015245	0.00426	0
384	0.771779	0.763475	0.754989	0.018131	0.010413	0
768	0.776446	0.78459	0.750546	0.026483	0.01459	0

Tabel 4.2 menunjukkan hasil evaluasi DBI serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan DBI dan standar deviasi diantara ketiga algoritma tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.2 dan Gambar 4.3.



Gambar 4.2 Grafik Invers DBI *Dataset* 2 Atribut – 2 *Cluster*

Berdasarkan grafik yang ditampilkan di Gambar 4.2 dapat dilihat bahwa algoritma HA-EKM memiliki kinerja yang baik saat diimplementasikan pada *dataset* 2 atribut – 2 *cluster* pada data yang berjumlah ≥ 384 data. Artinya untuk *clustering dataset* 2 atribut yang berjumlah data ≥ 384 data, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi DBI-nya yang dapat dilihat di Gambar 4.3.



Gambar 4.3 Grafik Standar Deviasi DBI *Dataset* 2 Atribut – 2 *Cluster*

Berdasarkan grafik yang ditampilkan di Gambar 4.3 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* 2 atribut – 2 *cluster*, algoritma HA-EKM memiliki standar deviasi yang selalu bernilai 0. Artinya hasil *clustering* algoritma HA-EKM lebih presisi dibandingkan dengan hasil *clustering* algoritma HA-KM dan HA-KPSO.

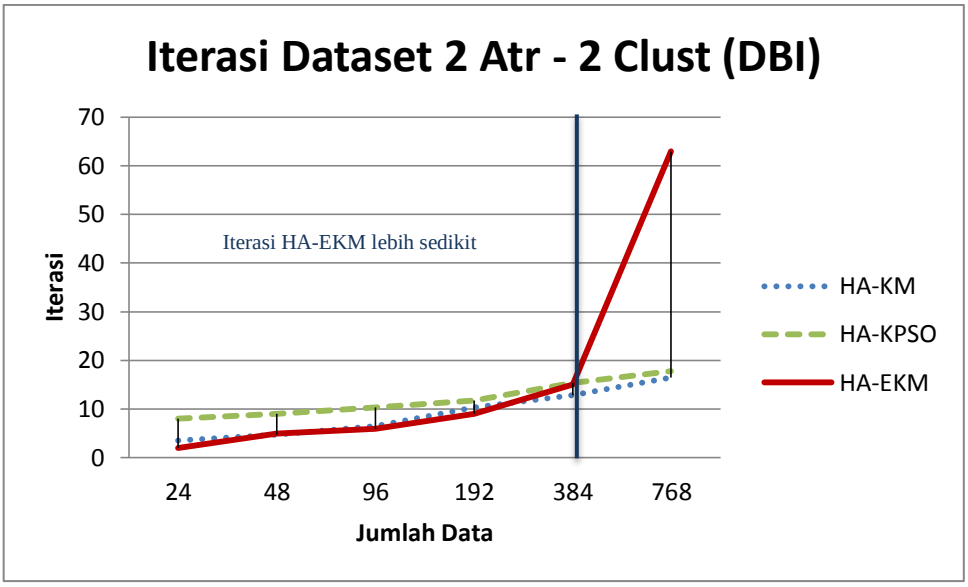
Untuk mengetahui *cost* yang dibutuhkan algoritma dalam menghasilkan hasil *clustering*, dilakukan perhitungan jumlah perulangan (iterasi). Jumlah iterasi algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* 2 atribut – 2 *cluster* disajikan di Tabel 4.3.

Tabel 4.3 Iterasi dan Standar Deviasi *Dataset* 2 Atribut – 2 *Cluster*

Jumlah data	Iterasi			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	3.5	8	2	0.971825	0	0
48	4.7	9	5	1.159502	0	0
96	6.5	10.3	6	2.223611	0.483046	0
192	10.3	11.8	9	4.110961	0.632456	0
384	12.9	15.4	15	3.754997	0.966092	0
768	16.5	17.8	63	7.276293	1.22927	0

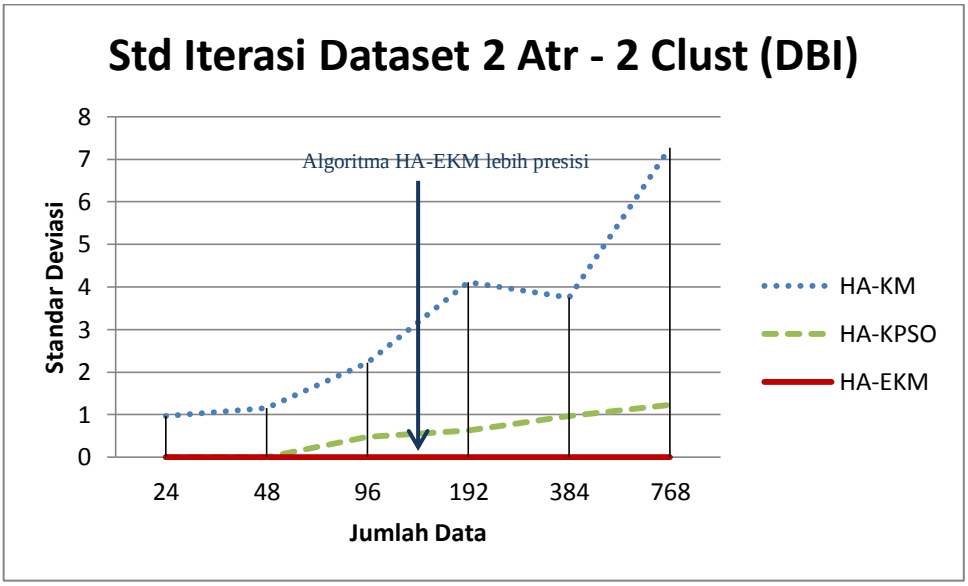
Tabel 4.3 menunjukkan jumlah iterasi serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan jumlah iterasi dan standar deviasi diantara ketiga

algoritma tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.4 dan Gambar 4.5.



Gambar 4.4 Grafik Iterasi *Dataset* 2 Atribut – 2 *Cluster* (DBI)

Berdasarkan grafik yang ditampilkan di Gambar 4.4 dapat dilihat bahwa algoritma HA-EKM memiliki jumlah iterasi sedikit saat diimplementasikan pada *dataset* 2 atribut – 2 *cluster* pada data yang berjumlah < 384 data. Sementara saat diimplementasikan pada *dataset* di atas 384 data, algoritma HA-EKM memiliki jumlah iterasi yang lebih banyak dibandingkan dengan algoritma HA-KM dan HA-KPSO. Jika dihubungkan dengan nilai DBI-nya terlihat bahwa pada *dataset* 2 atribut – 2 *cluster* semakin baik DBI algoritma HA-EKM maka semakin banyak juga jumlah iterasi yang diperlukan untuk menghasilkan hasil *clustering*. Artinya pada *dataset* 2 atribut yang berjumlah ≥ 384 data, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO karena dapat membentuk hasil *clustering* yang memiliki nilai DBI yang lebih baik dibandingkan dua algoritma lainnya, namun jumlah iterasi yang diperlukan untuk membentuk hasil *clustering* lebih banyak dibandingkan dua algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi iterasinya yang dapat dilihat di Gambar 4.5.



Gambar 4.5 Grafik Standar Deviasi Iterasi *Dataset* 2 Atribut – 2 *Cluster* (DBI)

Berdasarkan grafik yang ditampilkan di Gambar 4.5 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* 2 atribut – 2 *cluster*, algoritma HA-EKM memiliki standar deviasi iterasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

B. Hasil Pengujian *Dataset* 3 Atribut

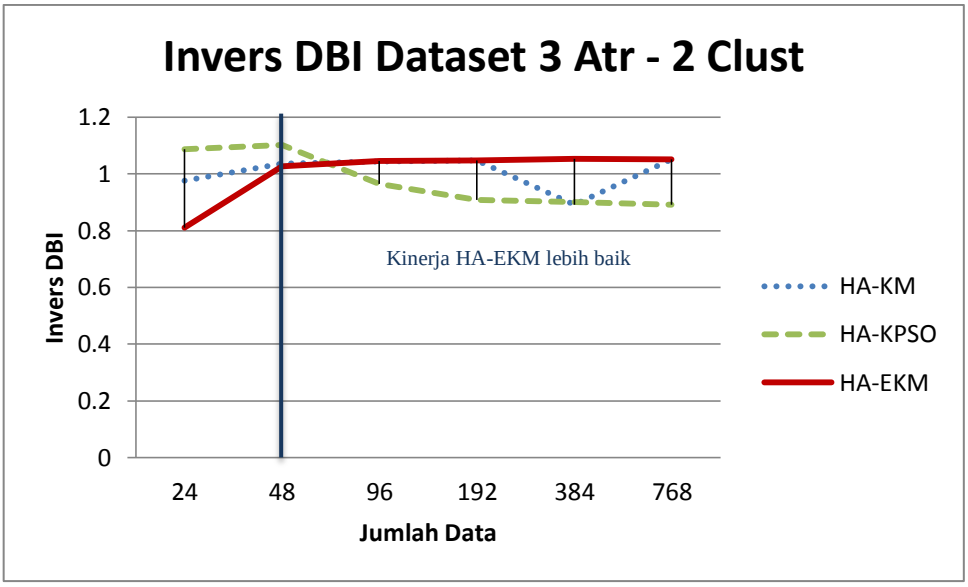
Evaluasi hasil *clustering* DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* 3 atribut – 2 *cluster* dapat dilihat di Tabel 4.4, sementara grafik kinerjanya dapat dilihat di Gambar 4.6 dan Gambar 4.7 untuk standar deviasinya.

Tabel 4.4 Nilai DBI dan Standar Deviasi *Dataset* 3 Atribut – 2 *Cluster*

Jumlah data	DBI			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	1.024857	0.920631	1.234757	0.171864	1.17E-16	0
48	0.965079	0.907822	0.973176	0.003604	0.022252	0
96	0.957379	1.036055	0.95599	0.001196	0.025965	0
192	0.955198	1.099998	0.955137	5.3E-05	0.070814	0
384	1.12233	1.110991	0.950226	0.544179	0.049942	0
768	0.952051	1.122307	0.95207	0.026483	0.04189	0

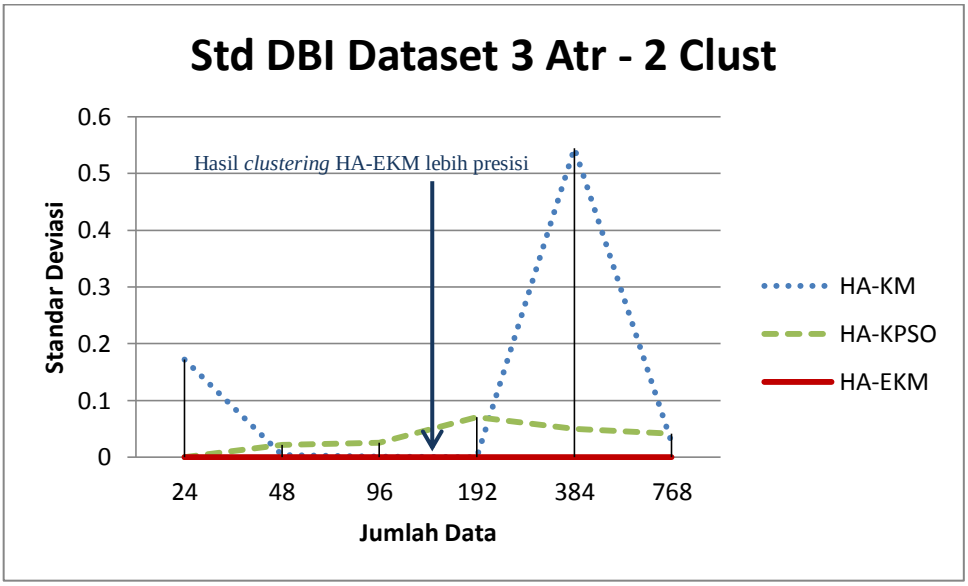
Tabel 4.4 menunjukkan hasil evaluasi DBI serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan DBI dan standar deviasi diantara ketiga algoritma

tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Grafik Invers DBI *Dataset* 3 Atribut – 2 *Cluster*

Berdasarkan grafik yang ditampilkan di Gambar 4.6 dapat dilihat bahwa algoritma HA-EKM memiliki kinerja yang baik saat diimplementasikan pada *dataset* 3 atribut – 2 *cluster* pada data yang berjumlah > 48 data. Artinya untuk *clustering dataset* 3 atribut yang berjumlah data > 48, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO karena memiliki hasil *clustering* dengan nilai DBI yang lebih baik dibandingkan dua algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi DBI-nya yang dapat dilihat di Gambar 4.7.



Gambar 4.7 Grafik Standar Deviasi DBI *Dataset* 3 Atribut – 2 *Cluster*

Berdasarkan grafik yang ditampilkan di Gambar 4.7 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* 3 atribut – 2 *cluster*, algoritma HA-EKM memiliki standar deviasi yang selalu bernilai 0. Artinya hasil *clustering* algoritma HA-EKM lebih presisi dibandingkan dengan hasil *clustering* algoritma HA-KM dan HA-KPSO.

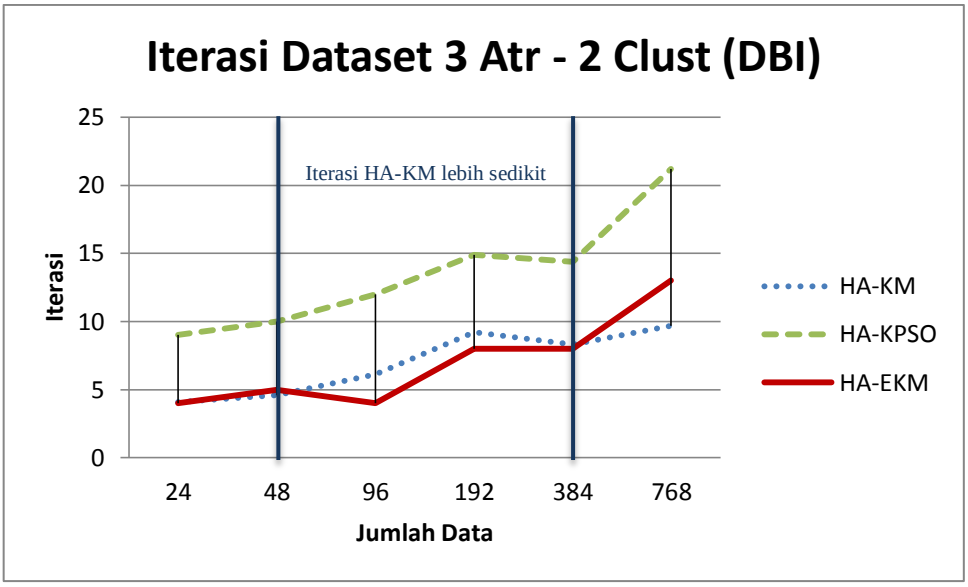
Untuk mengetahui *cost* yang dibutuhkan algoritma dalam menghasilkan hasil *clustering*, dilakukan perhitungan jumlah perulangan (iterasi). Jumlah iterasi algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* 3 atribut – 2 *cluster* disajikan di Tabel 4.5.

Tabel 4.5 Iterasi dan Standar Deviasi *Dataset* 3 Atribut – 2 *Cluster*

Jumlah data	Iterasi			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	4.1	9	4	1.911951	0	0
48	4.6	10	5	1.349897	0	0
96	6.1	12	4	2.558211	0.816497	0
192	9.2	14.9	8	2.936362	1.286684	0
384	8.3	14.4	8	2.626785	1.264911	0
768	9.7	21.2	13	2.263233	1.373961	0

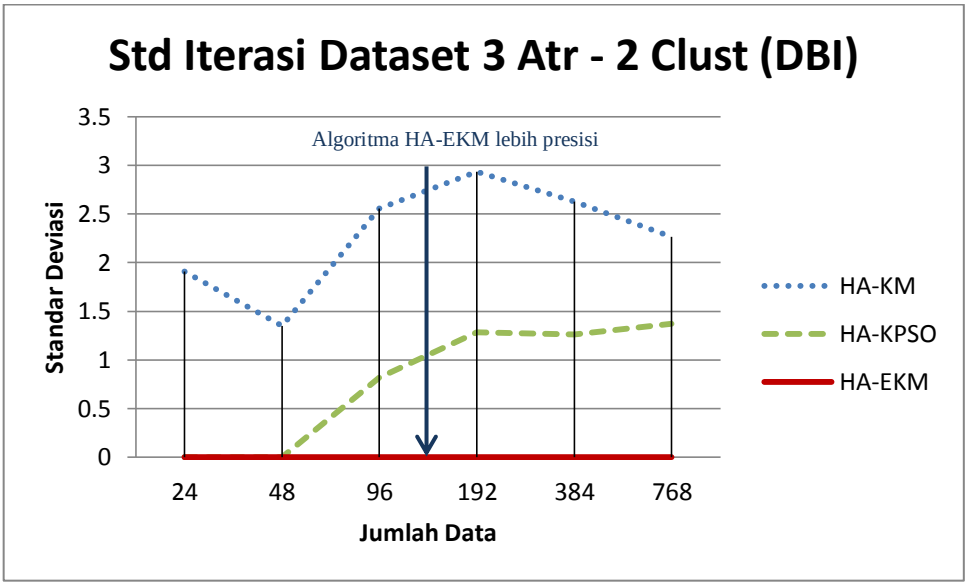
Tabel 4.5 menunjukkan jumlah iterasi serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan jumlah iterasi dan standar deviasi diantara ketiga

algoritma tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.8 dan Gambar 4.9.



Gambar 4.8 Grafik Iterasi *Dataset* 3 Atrbut – 2 *Cluster* (DBI)

Berdasarkan grafik yang ditampilkan di Gambar 4.8 dapat dilihat bahwa algoritma HA-EKM memiliki jumlah iterasi sedikit saat diimplementasikan pada *dataset* 3 atribut – 2 *cluster* pada data yang berjumlah > 48 dan ≤ 384 data. Sementara saat diimplementasikan pada *dataset* di atas 384 data, algoritma HA-EKM memiliki jumlah iterasi yang lebih banyak dibandingkan dengan algoritma HA-KM namun lebih sedikit saat dibandingkan dengan HA-KPSO. Jika dihubungkan dengan nilai DBI-nya terlihat bahwa pada *dataset* 3 atribut – 2 *cluster* semakin baik DBI algoritma HA-EKM maka semakin banyak juga iterasi yang diperlukan untuk menghasilkan hasil *clustering*. Artinya pada *dataset* 3 atribut yang berjumlah > 48 data, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO karena dapat membentuk hasil *clustering* yang memiliki nilai DBI yang lebih baik daripada dua algoritma lainnya dengan iterasi yang lebih sedikit daripada algoritma lainnya namun saat datanya > 384, jumlah iterasi yang dibutuhkan algoritma HA-EKM lebih banyak dibandingkan dengan algoritma HA-KM. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi iterasinya yang dapat dilihat di Gambar 4.9.



Gambar 4.9 Grafik Standar Deviasi Iterasi *Dataset* 3 Atribut – 2 *Cluster* (DBI)

Berdasarkan grafik yang ditampilkan di Gambar 4.9 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* 3 atribut – 2 *cluster*, algoritma HA-EKM memiliki standar deviasi iterasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

C. Hasil Pengujian *Dataset* 4 Atribut

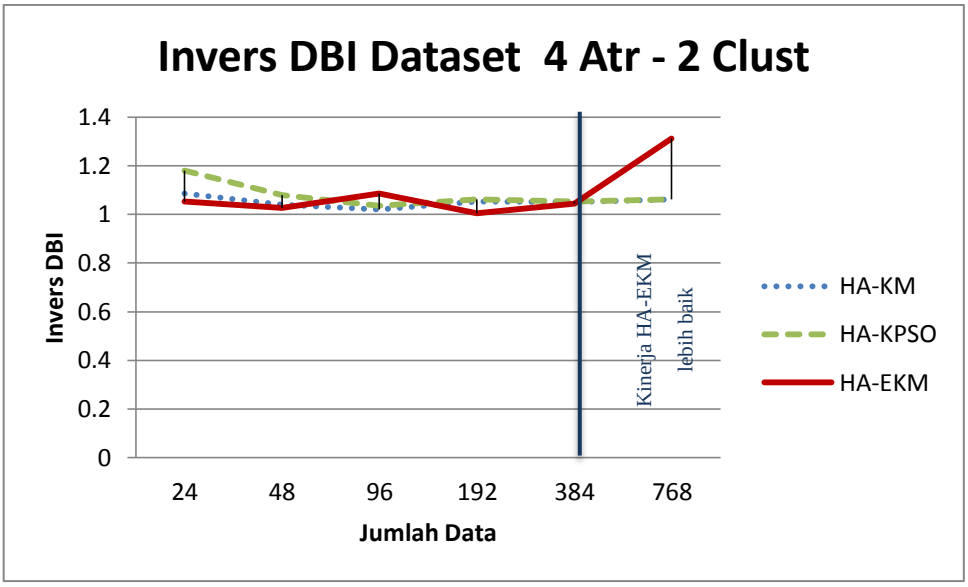
Evaluasi hasil *clustering* DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* 4 atribut – 2 *cluster* dapat dilihat di Tabel 4.6, sementara grafik kinerjanya dapat dilihat di Gambar 4.10 dan Gambar 4.11 untuk standar deviasinya.

Tabel 4.6 Nilai DBI dan Standar Deviasi *Dataset* 4 Atribut – 2 *Cluster*

Jumlah data	DBI			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	0.920382	0.847217	0.95004	0.095674	1.17E-16	0
48	0.960603	0.926229	0.972895	0.031372	0.005527	0
96	0.97971	0.965162	0.920317	0.019001	0.001878	0
192	0.948854	0.941397	0.994625	0.013605	0.01312	0
384	0.951551	0.949148	0.957704	0.014748	0.01127	0
768	0.941652	0.942066	0.761709	0.002511	0.002298	0

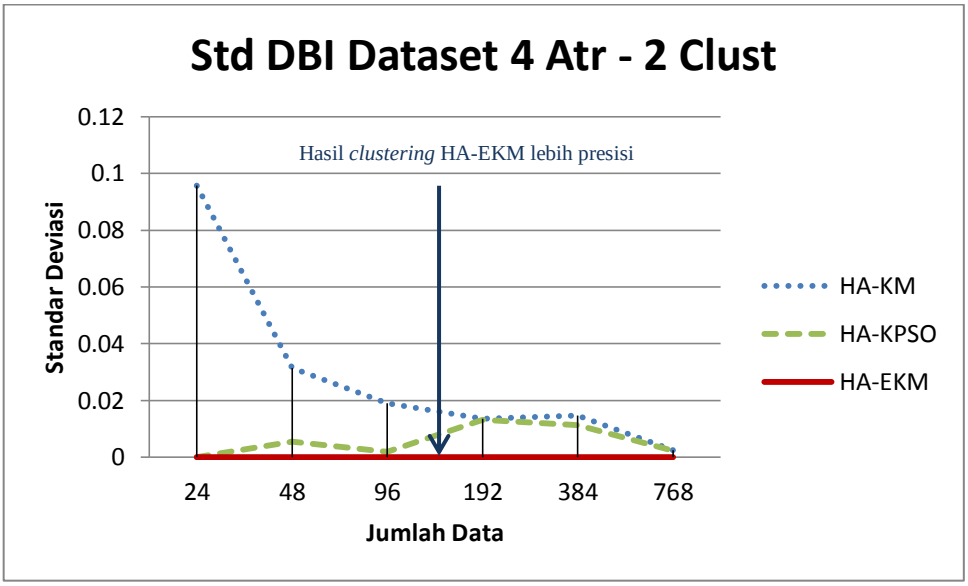
Tabel 4.6 menunjukkan hasil evaluasi DBI serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan DBI dan standar deviasi diantara ketiga algoritma

tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.10 dan Gambar 4.11.



Gambar 4.10 Grafik Invers DBI *Dataset* 4 Atribut – 2 *Cluster*

Berdasarkan grafik yang ditampilkan di Gambar 4.10 dapat dilihat bahwa algoritma HA-EKM memiliki kinerja yang baik saat diimplementasikan pada *dataset* 4 atribut – 2 *cluster* pada data yang berjumlah 768 data. Artinya untuk *clustering dataset* 4 atribut yang berjumlah di atas 768 data, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO karena memiliki hasil *clustering* dengan nilai DBI yang lebih baik dibandingkan dua algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi DBI-nya yang dapat dilihat di Gambar 4.11.



Gambar 4.11 Grafik Standar Deviasi DBI *Dataset* 4 Atribut – 2 *Cluster*

Berdasarkan grafik yang ditampilkan di Gambar 4.11 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* 4 atribut – 2 *cluster*, algoritma HA-EKM memiliki standar deviasi yang selalu bernilai 0. Artinya hasil *clustering* algoritma HA-EKM lebih presisi dibandingkan dengan hasil *clustering* algoritma HA-KM dan HA-KPSO.

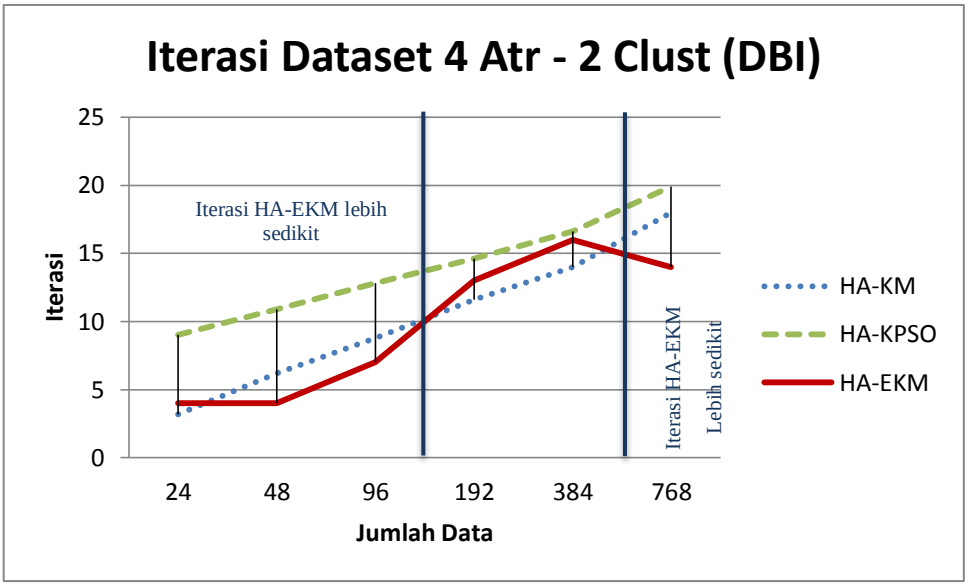
Untuk mengetahui *cost* yang dibutuhkan algoritma dalam menghasilkan hasil *clustering*, dilakukan perhitungan jumlah perulangan (iterasi). Jumlah iterasi algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* 4 atribut – 2 *cluster* disajikan di Tabel 4.7.

Tabel 4.7 Iterasi dan Standar Deviasi *Dataset* 4 Atribut – 2 *Cluster*

Jumlah data	Iterasi			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	3.2	9	4	0.788811	0	0
48	6.2	10.9	4	1.873796	0.316228	0
96	8.8	12.8	7	2.440401	0.632456	0
192	11.6	14.6	13	3.533962	0.699206	0
384	14	16.6	16	4.189935	1.074968	0
768	18	19.9	14	9.261629	1.37032	0

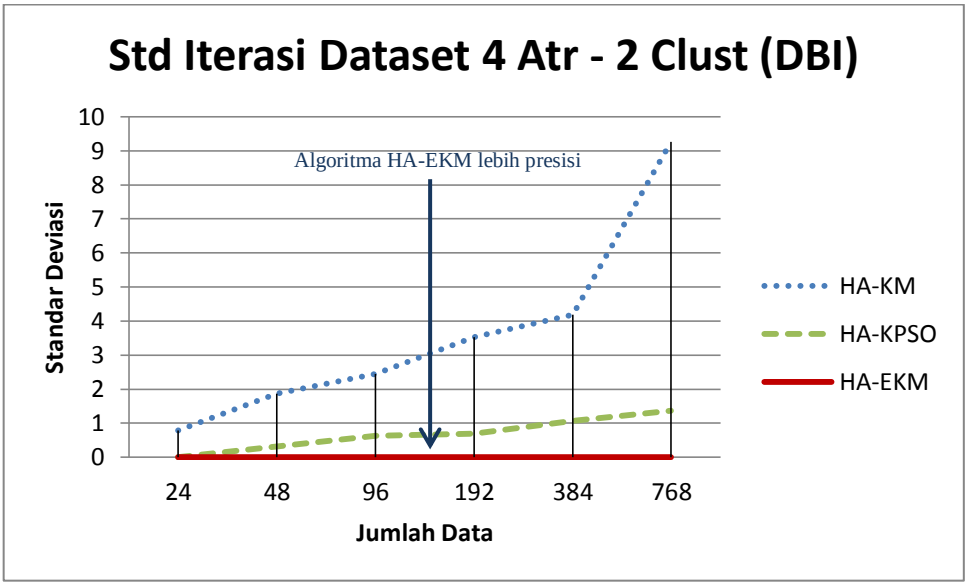
Tabel 4.7 menunjukkan jumlah iterasi serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan iterasi dan standar deviasi diantara ketiga algoritma

tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.12 dan Gambar 4.13.



Gambar 4.12 Grafik Iterasi *Dataset* 4 Atribut – 2 *Cluster* (DBI)

Berdasarkan grafik yang ditampilkan di Gambar 4.12 dapat dilihat bahwa algoritma HA-EKM memiliki jumlah iterasi yang lebih sedikit saat diimplementasikan pada *dataset* 4 atribut – 2 *cluster* pada data yang berjumlah < 192 data dan 768 data. Sementara saat diimplementasikan pada *dataset* yang berjumlah 192 dan 384 data, algoritma HA-EKM memiliki jumlah iterasi yang lebih banyak dari algoritma HA-KM namun lebih sedikit dari HA-KPSO. Jika dihubungkan dengan nilai DBI-nya terlihat bahwa pada *dataset* 4 atribut – 2 *cluster*, semakin banyak data maka semakin baik hasil DBI algoritma HA-EKM dengan jumlah iterasi yang lebih sedikit dibandingkan dengan algoritma HA-KM dan HA-KPSO. Artinya pada *dataset* 4 atribut yang berjumlah ≥ 768 data, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO karena dapat membentuk hasil *clustering* yang memiliki nilai DBI yang lebih baik dengan jumlah iterasi yang lebih sedikit daripada dua algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi iterasinya yang dapat dilihat di Gambar 4.13.



Gambar 4.13 Grafik Standar Deviasi Iterasi *Dataset* 4 Atribut – 2 *Cluster* (DBI)

Berdasarkan grafik yang ditampilkan di Gambar 4.13 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* 4 atribut – 2 *cluster*, algoritma HA-EKM memiliki standar deviasi iterasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

4.1.2 Hasil Pengujian Data Testing

Pengujian data testing dilakukan dengan menggunakan evaluasi DBI terhadap hasil *clustering* algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Iris, Wine dan Diabetes.

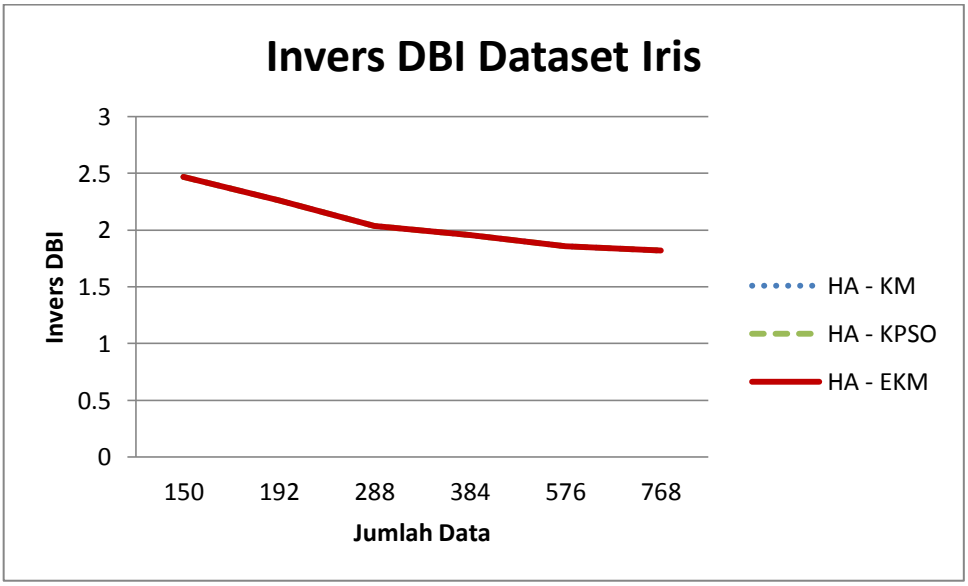
A. Hasil Pengujian Algoritma pada *Dataset* Iris

Evaluasi hasil *clustering* DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Iris dapat dilihat di Tabel 4.8, sementara grafik kinerjanya dapat dilihat di Gambar 4.14 dan Gambar 4.15 untuk standar deviasinya.

Tabel 4.8 Nilai DBI dan Standar Deviasi *Dataset* Iris

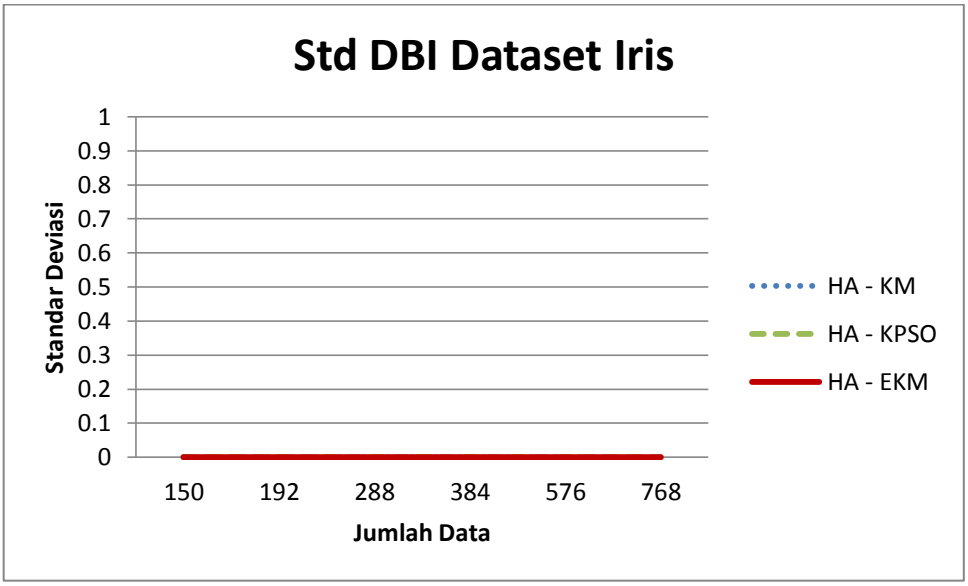
Jumlah data	DBI			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
150	0.404834	0.404834	0.404834	0	0	0
192	0.442028	0.442028	0.442028	0	0	0
288	0.49152	0.49152	0.49152	0	0	0
384	0.511712	0.511712	0.511712	0	0	0
576	0.538959	0.538961	0.538959	0	0	0
768	0.549095	0.549095	0.549095	0	0	0

Tabel 4.8 menunjukkan hasil evaluasi DBI serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan DBI dan standar deviasi diantara ketiga algoritma tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.14 dan Gambar 4.15.



Gambar 4.14 Grafik Invers DBI *Dataset* Iris

Berdasarkan grafik yang ditampilkan di Gambar 4.14 dapat dilihat bahwa algoritma HA-EKM saat diimplementasikan pada *dataset* Iris memiliki kinerja yang sama dengan algoritma HA-KM dan HA-KPSO untuk jumlah data mulai dari 150 hingga 768. Artinya hasil *clustering* yang dihasilkan ketiga algoritma adalah sama. Hal ini disebabkan karena dataset Iris memiliki data yang sudah terkelompok dengan baik seperti pada data model yang *cluster*nya sudah terpisah. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi DBI-nya yang dapat dilihat di Gambar 4.3.



Gambar 4.15 Grafik Standar Deviasi DBI *Dataset* Iris

Berdasarkan grafik yang ditampilkan di Gambar 4.15 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* Iris, baik algoritma HA-KM, HA-EKM dan HA-KPSO sama-sama memiliki standar deviasi 0. Artinya ketiga algoritma memiliki hasil *clustering* dengan presisi yang sama baik.

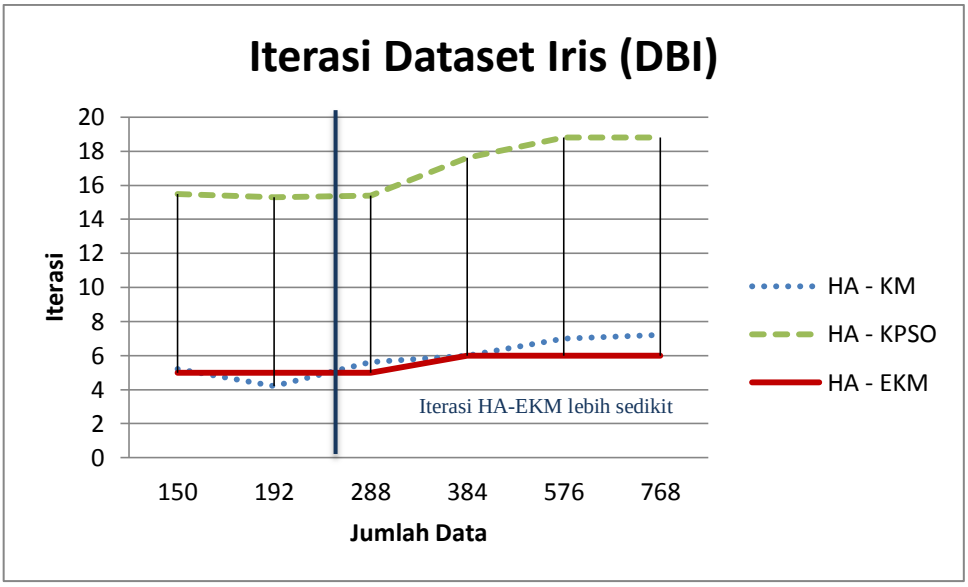
Untuk mengetahui *cost* yang dibutuhkan algoritma dalam menghasilkan hasil *clustering*, dilakukan perhitungan jumlah perulangan (iterasi). Jumlah iterasi algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Iris disajikan di Tabel 4.9.

Tabel 4.9 Iterasi dan Standar Deviasi *Dataset* Iris (DBI)

Jumlah data	Iterasi			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
150	5.2	15.5	5	0.918937	0.707107	0
192	4.2	15.3	5	1.813529	1.159502	0
288	5.6	15.4	5	1.349897	1.264911	0
384	6	17.6	6	1.333333	1.173788	0
576	7	18.8	6	1.699673	1.988858	0
768	7.2	18.8	6	1.75119	2.097618	0

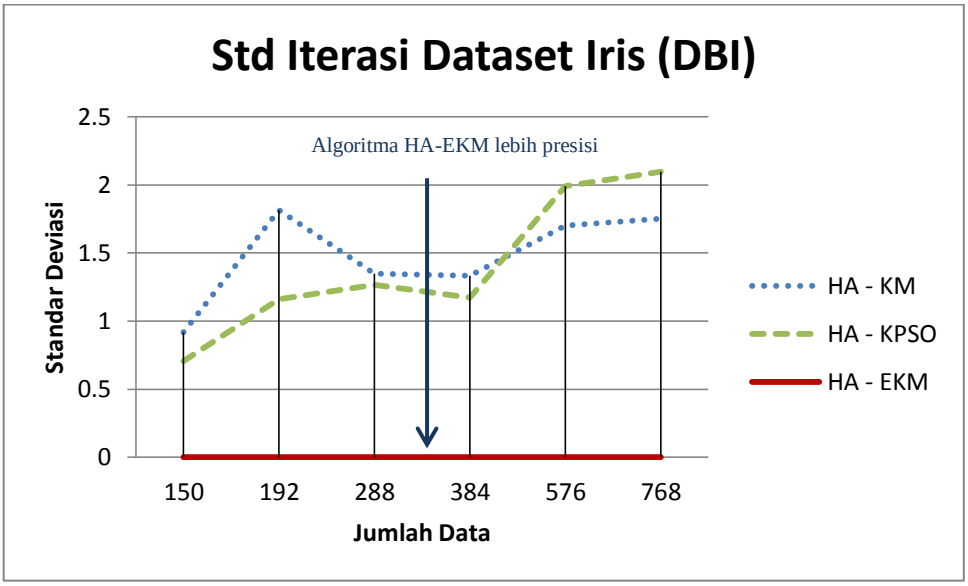
Tabel 4.9 menunjukkan jumlah iterasi serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan iterasi dan standar deviasi diantara ketiga algoritma

tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.16 dan Gambar 4.17.



Gambar 4.16 Grafik Iterasi *Dataset* Iris

Berdasarkan grafik yang ditampilkan di Gambar 4.16 dapat dilihat bahwa algoritma HA-EKM memiliki jumlah iterasi yang cenderung lebih sedikit dari algoritma HA-KM dan HA-EKM. Artinya pada dataset Iris, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO sebab dengan nilai DBI yang sama, algoritma HA-EKM memerlukan jumlah iterasi yang lebih sedikit dibandingkan dua algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi iterasinya yang dapat dilihat di Gambar 4.17.



Gambar 4.17 Grafik Standar Deviasi Iterasi *Dataset* Iris

Berdasarkan grafik yang ditampilkan di Gambar 4.17 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* Iris, algoritma HA-EKM memiliki standar deviasi iterasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

B. Hasil Pengujian Algoritma pada *Dataset* Wine

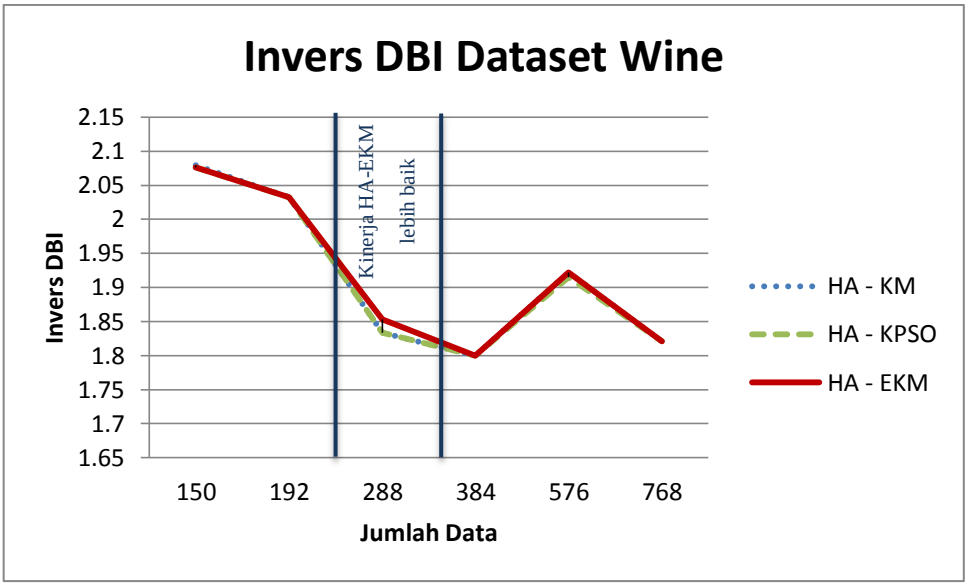
Evaluasi hasil *clustering* DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Wine dapat dilihat di Tabel 4.10, sementara grafik kinerjanya dapat dilihat di Gambar 4.18 dan Gambar 4.19 untuk standar deviasinya.

Tabel 4.10 Nilai DBI dan Standar Deviasi *Dataset* Wine

Jumlah data	DBI			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
150	0.480827	0.481703	0.481703	0.00141	0	5.85E-17
192	0.491931	0.491931	0.491931	5.85E-17	0	5.85E-17
288	0.545498	0.545513	0.539592	0.01245	0.007406	0
384	0.555552	0.555552	0.555552	1.17E-16	0	1.17E-16
576	0.520239	0.522243	0.520257	1.94E-05	0.002563	0
768	0.549197	0.549197	0.549197	1.17E-16	0	1.17E-16

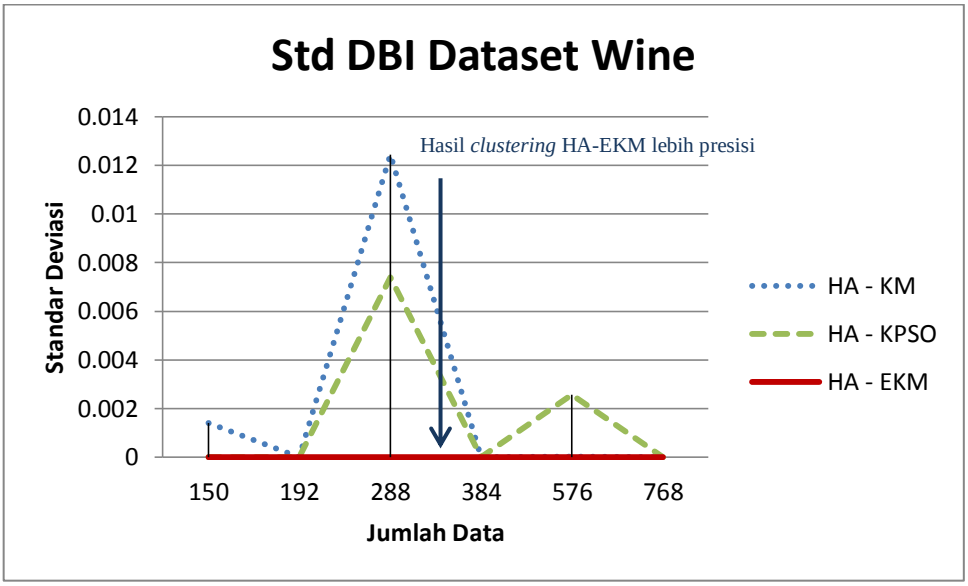
Tabel 4.10 menunjukkan hasil evaluasi DBI serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan DBI dan standar deviasi diantara ketiga algoritma

tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.18 dan Gambar 4.19.



Gambar 4.18 Grafik Invers DBI *Dataset* Wine

Berdasarkan grafik yang ditampilkan di Gambar 4.18 dapat dilihat bahwa algoritma HA-EKM saat diimplementasikan pada *dataset* Wine memiliki kinerja yang hampir sama dengan algoritma HA-KM dan HA-KPSO, hanya saja pada jumlah data 288 algoritma HA-EKM memiliki kinerja yang lebih unggul dibandingkan dengan dua algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi DBI-nya yang dapat dilihat di Gambar 4.19.



Gambar 4.19 Grafik Standar Deviasi DBI *Dataset* Wine

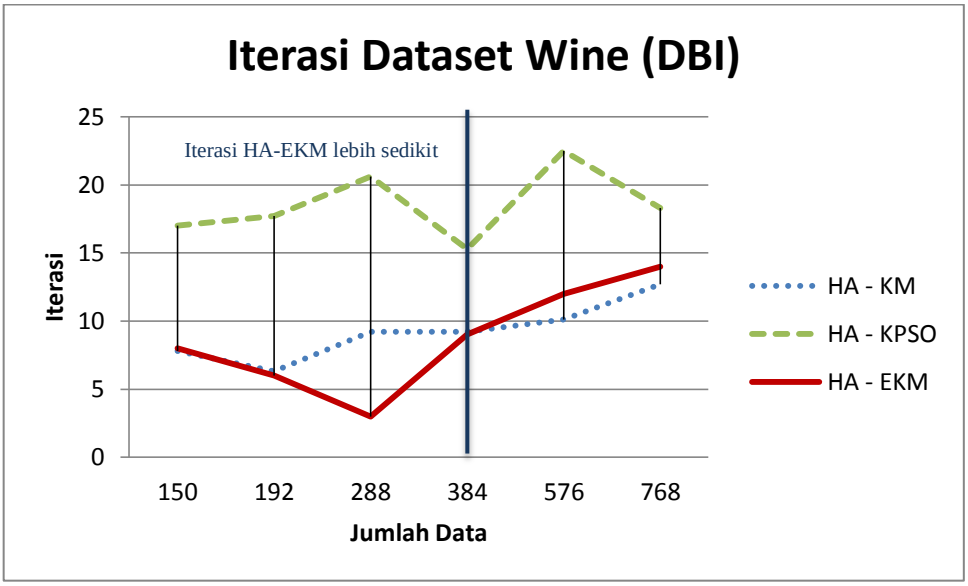
Berdasarkan grafik yang ditampilkan di Gambar 4.19 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* Wine, algoritma HA-EKM memiliki standar deviasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

Untuk mengetahui *cost* yang dibutuhkan algoritma dalam menghasilkan hasil *clustering*, dilakukan perhitungan jumlah perulangan (iterasi). Iterasi algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Wine disajikan di Tabel 4.11.

Tabel 4.11 Iterasi dan Standar Deviasi *Dataset* Wine

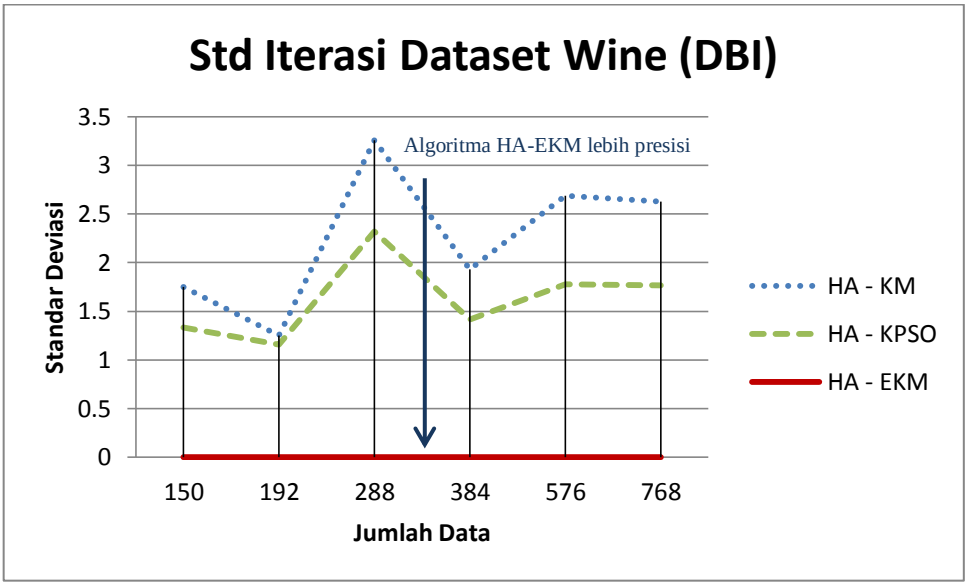
Jumlah data	Iterasi			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
150	7.8	17	8	1.75119	1.333333	0
192	6.3	17.7	6	1.251666	1.159502	0
288	9.2	20.6	3	3.259175	2.319004	0
384	9.2	15.3	9	1.932184	1.418136	0
576	10.1	22.5	12	2.685351	1.779513	0
768	12.7	18.3	14	2.626785	1.766981	0

Tabel 4.11 menunjukkan jumlah iterasi serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan iterasi dan standar deviasi diantara ketiga algoritma tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.20 dan Gambar 4.21.



Gambar 4.20 Grafik Iterasi *Dataset* Wine

Berdasarkan grafik yang ditampilkan di Gambar 4.20 dapat dilihat bahwa saat diimplementasikan pada *dataset* Wine algoritma HA-EKM memiliki jumlah iterasi yang lebih sedikit dari algoritma HA-KM pada jumlah data < 384 data, sementara pada jumlah data ≥ 384 algoritma HA-EKM memiliki jumlah iterasi yang lebih banyak dari HA-KM. Akan tetapi jika dibandingkan dengan algoritma HA-KPSO, algoritma HA-EKM selalu memiliki jumlah iterasi yang lebih sedikit. Artinya pada *dataset* Wine, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO sebab dengan nilai DBI yang sama, algoritma HA-EKM memerlukan jumlah iterasi yang lebih sedikit dibandingkan algoritma HA-KPSO dan walaupun pada jumlah data > 384 jumlah iterasinya lebih banyak daripada algoritma HA-KM namun hasil *clustering* algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi iterasinya yang dapat dilihat di Gambar 4.21.



Gambar 4.21 Grafik Standar Deviasi Iterasi *Dataset* Wine

Berdasarkan grafik yang ditampilkan di Gambar 4.21 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* Wine, algoritma HA-EKM memiliki standar deviasi iterasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

C. Hasil Pengujian Algoritma pada *Dataset* Diabetes

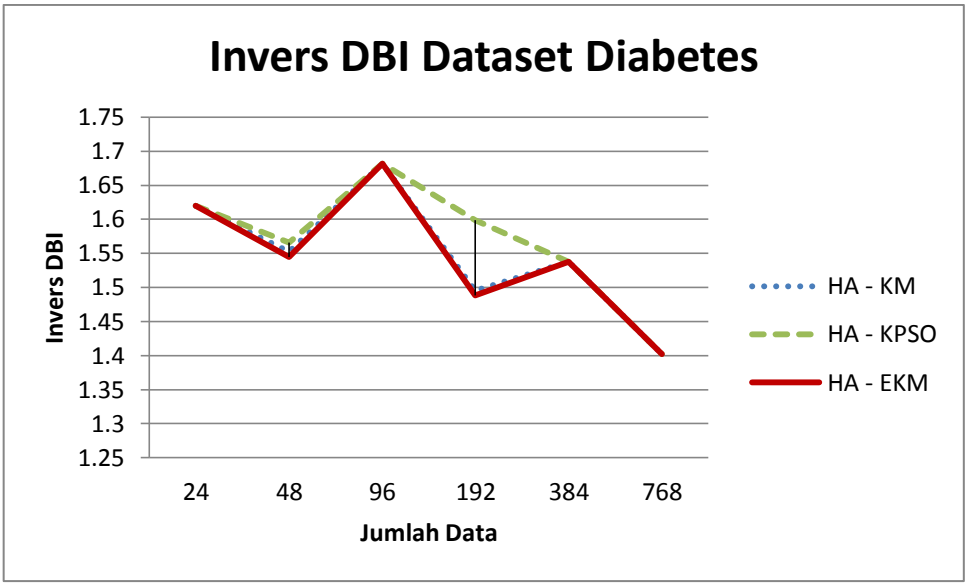
Evaluasi hasil *clustering* DBI untuk algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Diabetes dapat dilihat di Tabel 4.12, sementara grafik nya dapat dilihat di Gambar 4.22 dan Gambar 4.23 untuk standar deviasinya.

Tabel 4.12 Nilai DBI dan Standar Deviasi *Dataset* Diabetes

Jumlah data	DBI			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	0.617203	0.617203	0.617203	1.17E-16	0	0
48	0.644038	0.638737	0.647282	0.008954	0	0
96	0.959841	0.986539	1.031952	0.136464	0.040946	0
192	0.668794	0.625349	0.671797	0.009497	0	0
384	0.65033	0.65033	0.65033	1.17E-16	0	0
768	0.713382	0.713382	0.713382	1.17E-16	0	0

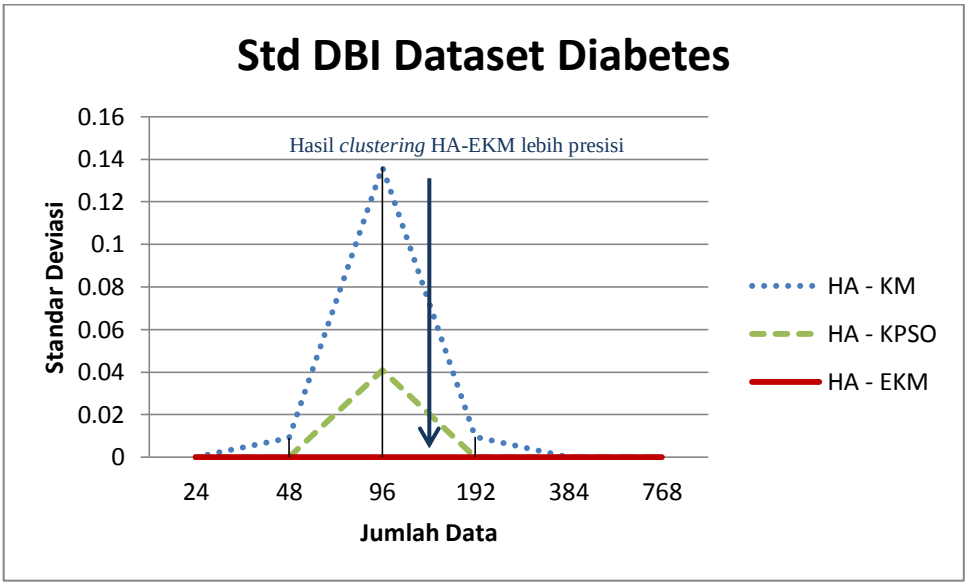
Tabel 4.12 menunjukkan hasil evaluasi DBI serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan DBI dan standar deviasi diantara ketiga algoritma

tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.22 dan Gambar 4.23.



Gambar 4.22 Grafik Invers DBI *Dataset* Diabetes

Berdasarkan grafik yang ditampilkan di Gambar 4.22 dapat dilihat bahwa saat diimplementasikan pada *dataset* Diabetes algoritma HA-EKM memiliki kinerja yang lebih buruk dibandingkan dengan algoritma HA-KM dan HA-KPSO pada data yang berjumlah < 384 data, sedangkan pada data yang berjumlah ≥ 384 data, algoritma HA-EKM memiliki kinerja yang sama dengan dua algoritma lainnya. Artinya algoritma HA-EKM baik diimplementasikan pada dataset diabetes pada data berjumlah ≥ 384 data. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi DBI-nya yang dapat dilihat di Gambar 4.23.



Gambar 4.23 Grafik Standar Deviasi DBI *Dataset* Diabetes

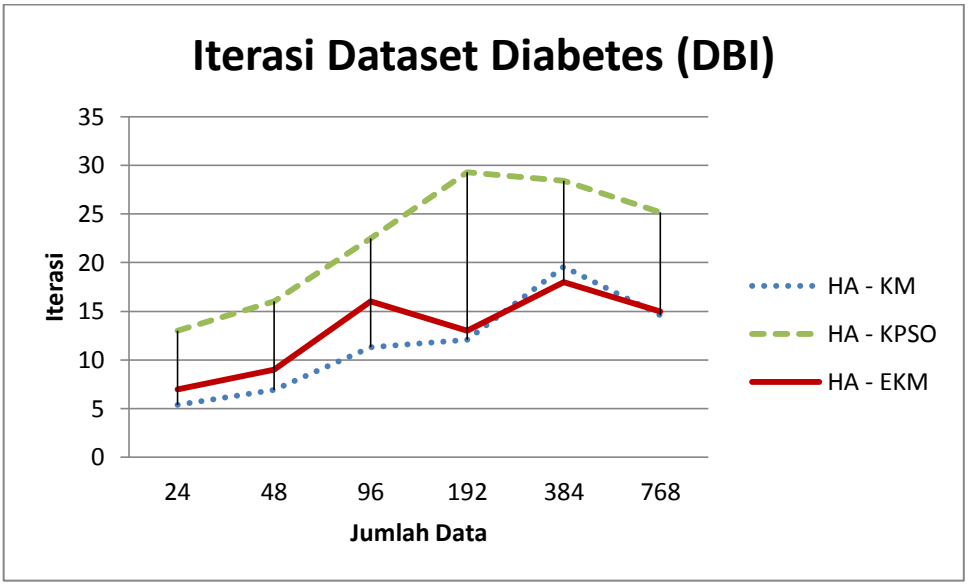
Berdasarkan grafik yang ditampilkan di Gambar 4.23 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* Diabetes, algoritma HA-EKM memiliki standar deviasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

Untuk mengetahui *cost* yang dibutuhkan algoritma dalam menghasilkan hasil *clustering*, dilakukan perhitungan jumlah perulangan (iterasi). Iterasi algoritma HA-KM, HA-KPSO dan HA-EKM pada *dataset* Diabetes disajikan di Tabel 4.13.

Tabel 4.13 Iterasi dan Standar Deviasi *Dataset* Diabetes

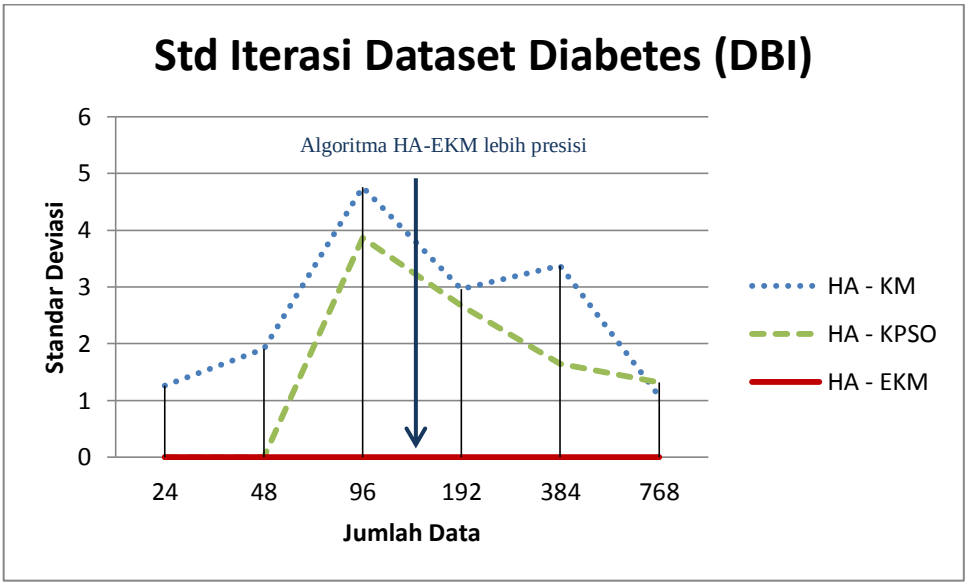
Jumlah data	Iterasi			Standar Deviasi		
	HA-KM	HA-KPSO	HA-EKM	HA-KM	HA-KPSO	HA-EKM
24	5.4	13	7	1.264911	0	0
48	6.9	16	9	1.911951	0	0
96	11.3	22.5	16	4.762119	3.865805	0
192	12.1	29.3	13	2.960856	2.668749	0
384	19.6	28.4	18	3.373096	1.646545	0
768	14.6	25.2	15	1.074968	1.316561	0

Tabel 4.13 menunjukkan jumlah iterasi serta standar deviasi untuk masing-masing algoritma HA-KM, HA-KPSO dan HA-EKM. Untuk membantu memudahkan perbandingan iterasi dan standar deviasi diantara ketiga algoritma tersebut maka hasil pengujian divisualisasikan ke dalam grafik yang ditampilkan di Gambar 4.24 dan Gambar 4.25.



Gambar 4.24 Grafik Iterasi *Dataset* Diabetes

Berdasarkan grafik yang ditampilkan di Gambar 4.24 dapat dilihat bahwa saat diimplementasikan pada *dataset* Diabetes, algoritma HA-EKM memiliki jumlah iterasi yang lebih banyak dibandingkan dengan algoritma HA-KM pada data dengan jumlah < 384. Sementara pada jumlah data ≥ 384 algoritma HA-EKM memiliki jumlah iterasi yang lebih sedikit dari HA-KM. Akan tetepi algoritma HA-EKM memiliki jumlah iterasi yang lebih sedikit dibandingkan dengan algoritma HA-KPSO. Artinya pada *dataset* Diabetes, algoritma HA-EKM lebih baik digunakan daripada algoritma HA-KM dan HA-KPSO pada data berjumlah ≥ 384, sebab dengan nilai DBI yang sama, algoritma HA-EKM memerlukan jumlah iterasi yang lebih sedikit dibandingkan algoritma HA-KPSO dan HA-KM. Selain itu hasil *clustering* algoritma HA-EKM lebih presisi dibandingkan dengan algoritma lainnya. Selanjutnya, untuk melihat kepresisian algoritma maka dilakukan perbandingan standar deviasi iterasinya yang dapat dilihat di Gambar 4.25.



Gambar 4.25 Grafik Standar Deviasi Iterasi *Dataset* Diabetes

Berdasarkan grafik yang ditampilkan di Gambar 4.25 dapat dilihat bahwa pada saat diimplementasikan pada *dataset* Diabetes, algoritma HA-EKM memiliki standar deviasi iterasi yang selalu bernilai 0. Artinya algoritma HA-EKM lebih presisi dibandingkan dengan algoritma HA-KM dan HA-KPSO.

4.1.3 Analisis Perbandingan Algoritma

Berdasarkan hasil pengujian yang telah dilakukan diketahui bahwa algoritma HA-EKM memiliki kinerja yang baik berdasarkan evaluasi *clustering* DBI pada *dataset* dengan jumlah atribut kurang dari sama dengan 4. Algoritma HA-EKM memiliki kinerja yang lebih baik dibandingkan dengan algoritma HA-KM dan HA-KPSO pada *dataset* dengan jumlah *cluster* kelas 2 sedangkan pada jumlah *cluster* kelas lebih dari 2, algoritma HA-EKM memiliki kinerja yang sama dengan algoritma HA-KM dan HA-KPSO. Hal ini terlihat baik pada *dataset* model maupun *dataset* testing. Pada *dataset* 2 *cluster*, algoritma HA-EKM menunjukkan kinerja yang lebih baik pada jumlah atribut kurang dari sama dengan 4 atribut, dengan jumlah data lebih dari sama dengan 768 data. Pada *dataset* yang datanya sudah membentuk *cluster* dengan baik seperti pada data testing, algoritma HA-EKM memiliki nilai DBI yang sama atau hampir sama dengan algoritma HA-KM dan HA-KPSO.

Algoritma HA-EKM memerlukan *cost* (jumlah iterasi) yang lebih sedikit dibandingkan dengan algoritma HA-KPSO sedangkan *cost* yang diperlukan algoritma HA-KM terkadang lebih banyak dan terkadang lebih sedikit dibandingkan dengan algoritma HA-EKM. Namun algoritma HA-EKM memiliki presisi yang paling baik jika dibandingkan dengan algortima HA-KM dan HA-KPSO.

4.2 Hasil Pengujian Data Kasus

Data lulusan mahasiswa yang sudah ditansformasi seperti yang dijelaskan di BAB 3 subbab 3.2.1 diolah menggunakan algoritma HA-EKM. Sebagai pembanding, data juga diolah menggunakan algoritma HA-KM dan HA-KPSO. Hasil *clustering* masing-masing algoritma kemudian dievaluasi menggunakan DBI. Hasil evaluasi DBI ketiga algoritma dapat dilihat di Tabel 4.14.

Tabel 4.14 Nilai DBI dan Iterasi Algoritma HA-EKM, HA-KM dan HA-KPSO

Algoritma	Jumlah *Cluster*	DBI		Iterasi	
		DBI	Standar Deviasi	Iterasi	Standar Deviasi
HA-EKM	6	0.796168	0	14	0
HA-KM	6	0.822642	0.040755	26	12.93574
HA-KPSO	6	0.808547	0.007049	29	2.880972

Berdasarkan hasil evaluasi yang disajikan di Tabel 4.14 terlihat bahwa algoritma HA-EKM menghasilkan nilai DBI yang 3.22 % lebih kecil dari algoritma HA-KM dan 1.53 % lebih kecil dari algoritma HA-KPSO. Hal tersebut menunjukkan bahwa hasil *clustering* algoritma HA-EKM lebih baik daripada hasil *clustering* HA-KM dan HA-KPSO. Jika dilihat berdasarkan nilai standar deviasi DBI dan jumlah iterasi, algoritma HA-EKM juga memiliki standar deviasi yang bernilai 0, sementara algoritma HA-KM bernilai lebih dari 0. Hal tersebut menunjukkan bahwa hasil *clustering* algoritma HA-EKM lebih presisi dibandingkan dengan hasil *clustering* algoritma HA-KM.