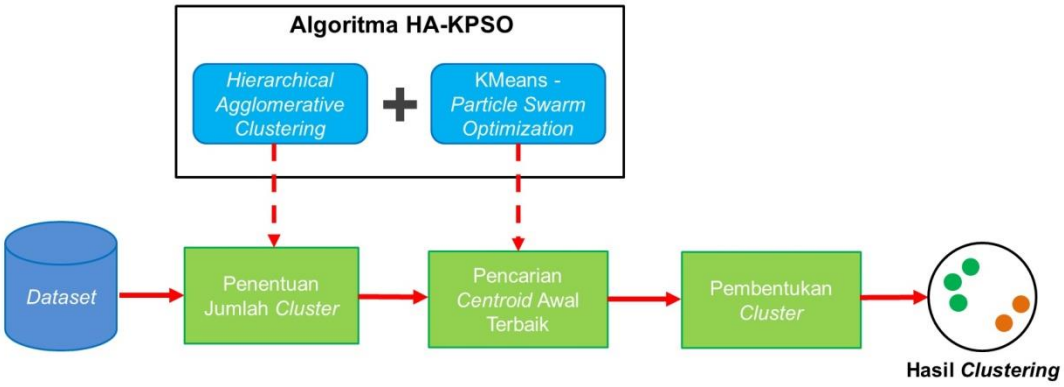
BAB 3
METODE PENELITIAN

3.1 *Hierarchical Agglomerative – KMeans Particle Swarm Optimization*

Penelitian ini mengusulkan algoritma *Hierarchical Agglomerative – KMeans Particle Swarm Optimization* (HA-KPSO), sebuah algoritma yang menggabungkan kelebihan dari algoritma *Hierarchical Agglomerative clustering* (HAC) dan *Kmeans-particle swarm optimization* (K-PSO) untuk menduga jumlah *cluster* sekaligus mencari pusat *centroid* awal terbaik dari suatu *dataset*.

3.2 Teknik Penggabungan Algoritma

Model dari algoritma HA-KPSO yang diusulkan terdiri dari tiga proses utama yang dapat dilihat di Gambar 3.1.

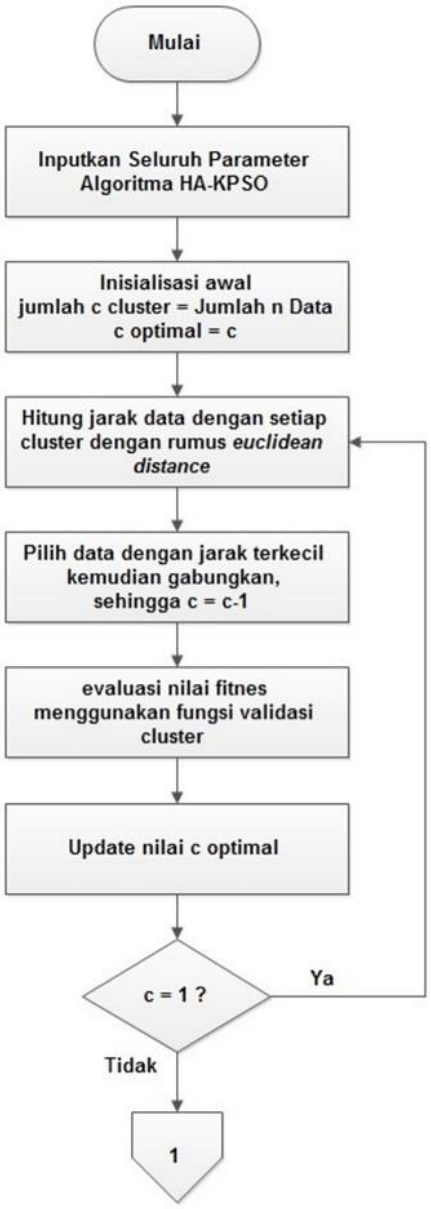


Gambar 3.1 Teknik Penggabungan Algoritma

Tahap pertama adalah proses menentukan jumlah *cluster* menggunakan algoritma *Hierarchical Agglomerative Clustering*. Selanjutnya adalah proses mencari *centroid* awal terbaik dari *dataset* menggunakan algoritma *Particle Swarm Optimization*. Tahap terakhir adalah proses pembentukan *cluster* dengan cara mengelompokan data ke dalam *cluster* yang sesuai menggunakan persamaan *euclidean distance*. Proses *clustering* menggunakan algoritma HA-KPSO dijelaskan lebih lanjut di subbab selanjutnya.

3.2.1 Proses Penentuan jumlah *Cluster*

Langkah awal dari algoritma HA-KPSO adalah menentukan jumlah *cluster* yang dapat dibentuk dari *dataset*. Algoritma HA-KPSO menggunakan konsep algoritma HAC yang dapat menduga jumlah *cluster* terbaik. Proses penentuan jumlah *cluster* dibagi menjadi beberapa langkah yang dapat dilihat pada flowchact di Gambar 3.2.

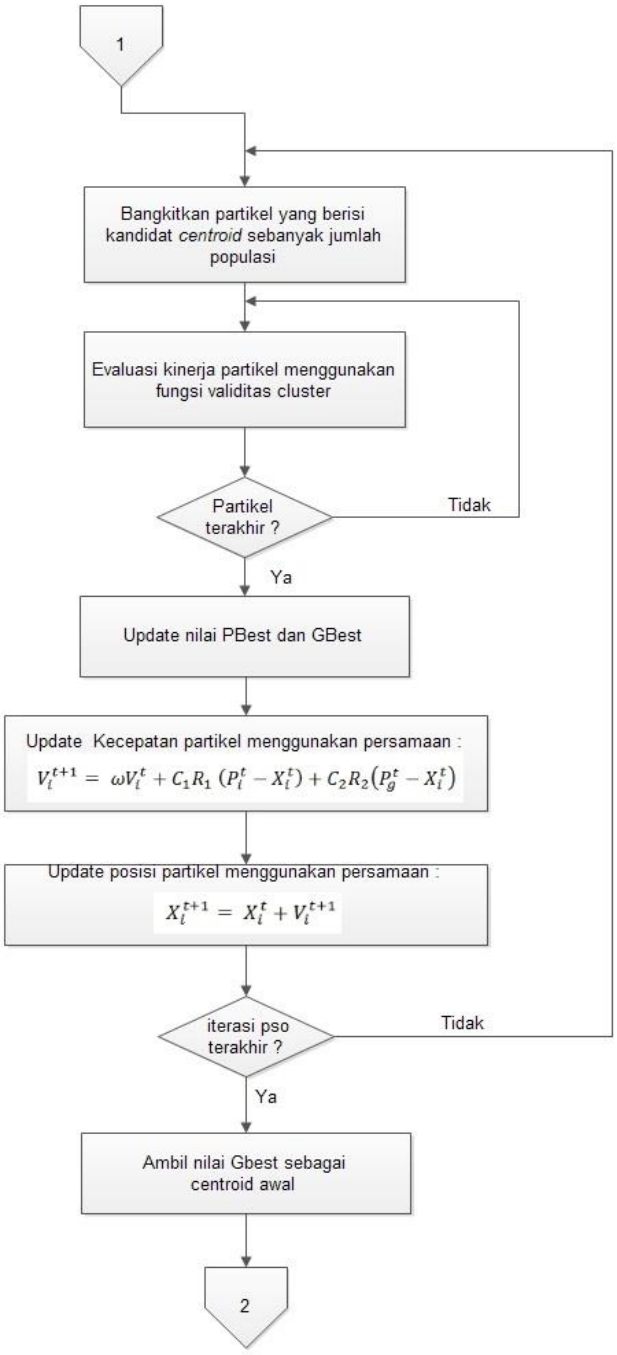


Gambar 3.2 *Flowchart* Proses Penentuan Jumlah *Cluster* Algoritma HA-KPSO

Pada Algoritma *Hierarchical Agglomerative Clustering*, sebelum dilakukan pengelompokan, langkah pertama adalah mengasumsikan setiap data sebagai *cluster*. Jika terdapat data sebanyak *n* dan *c* dianggap sebagai jumlah *cluster*, maka $n = c$. Di awal proses diasumsikan juga bahwa *c* optimal adalah *c*. Selanjutnya dilakukan perhitungan jarak dengan menggunakan persamaan *Euclidean distance* untuk mengetahui jarak dari masing-masing *cluster*. Berdasarkan jarak antar *cluster*, dipilih *cluster* yang memiliki jarak paling minimal dari seluruh *cluster* kemudian gabungkan sehingga $c = c - 1$. Proses akan terus berlanjut hingga $c = 1$. Pada setiap proses penggabungan *cluster*, dilakukan evaluasi hasil *clustering* menggunakan fungsi validasi *cluster*. Fungsi validasi *cluster* yang digunakan di penelitian ini selengkapnya dijelaskan di Subbab 2.8. Berdasarkan hasil evaluasi tersebut dilakukan peng-*update*-an nilai *c* optimal dengan aturan jika hasil evaluasi *cluster* lebih baik dari sebelum penggabungan, maka nilai $c\ opt = c$. Sedangkan jika hasil evaluasi *cluster* tidak lebih baik, maka nilai *c opt* tidak berubah. Dugaan jumlah cluster yang dihasilkan pada proses ini akan digunakan sebagai masukan jumlah cluster untuk setiap proses algoritma HA-KPSO selanjutnya.

3.2.2 Proses Pencarian *Centroid* Awal Terbaik

Langkah selanjutnya dari algoritma HA-KPSO adalah proses mencari *centroid* awal terbaik. algoritma HA-KPSO menggunakan konsep dari algoritma K-PSO yang dapat menemukan *centroid* terbaik dari suatu *dataset*. *Centroid* terbaik didapatkan dengan membangkitkan sekumpulan partikel, dimana setiap partikel merepresentasikan kandidat solusi yang berupa nilai *centroid*. Selanjutnya Fungsi optimasi dari PSO akan mencari partikel terbaik dari kumpulan partikel yang telah dibangkitkan tersebut. Partikel terbaik yang ditemukan akan digunakan sebagai nilai *centroid* awal yang digunakan untuk proses clustering algoritma HA-KPSO. Proses pencarian *centroid* awal terbaik dibagi ke dalam beberapa langkah yang dapat dilihat pada *flowchart* di Gambar 3.3.

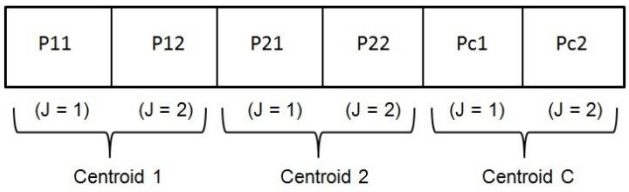


Gambar 3.3 *Flowchart* Proses Pencarian *Centroid* Awal Terbaik Algoritma HA-KPSO

1. Inisialisasi Posisi dan Kecepatan Awal Partikel

Langkah pertama adalah melakukan inisialisasi nilai ke dalam partikel. Setiap partikel yang dibangkitkan pada penelitian ini merupakan representasi dari nilai *centroid* yang menjadi kandidat solusi. Setiap partikel pada algoritma HA-KPSO

dibentuk dari kumpulan *centroid* yang setiap *centroid* terdiri dari banyaknya atribut pada data. Apabila centroid dimisalkan sebagai c ={1, 2,….c} dan atribut dimisalkan sebagai j={1,2,…j} maka setiap partikel memiliki panjang P = c * j. Gambaran inisialisasi partikel pada penelitian ini ditunjukkan pada Gambar 3.4



Gambar 3.4 Representasi Partikel Pada Algoritma HA-KPSO

Pada Gambar 3.4, dua kolom pertama merupakan representasi *centroid* untuk *cluster* ke - 1, dua kolom selanjutnya menunjukan representasi *centroid* pada *cluster* ke - 2 dan seterusnya sampe *centroid* pada *cluster* ke - C. Setiap cluster terdiri dari atribut-atribut yang jumlah atributnya menyesuaikan dengan atribut pada dataset digunakan. Pada penelitian ini jumlah *Centroid* yang akan dibentuk oleh partikel menyesuikan dari jumlah *cluster* yang dihasilkan proses penentuan jumlah *cluster* sebelumnya. Misalkan jumlah *cluster* terbaik dari suatu kasus adalah tiga cluster maka jumlah *centroid* yang dibangkitkan pada partikel adalah tiga *centroid*.

Inisialisasi nilai partikel pada algoritma HA-KPSO diperoleh secara acak dari data didalam *dataset*. Sebagai contoh misalkan kita akan membuat partikel dari *dataset* yang memiliki 4 atribut dan jumlah *cluster* yang akan dibentuk adalah dua *cluster*, maka representasi partikel pada kasus ini dapat dilihat di Gambar 3.5 berikut.



Gambar 3.5 Representasi Partikel dengan Nilai *Centroid*

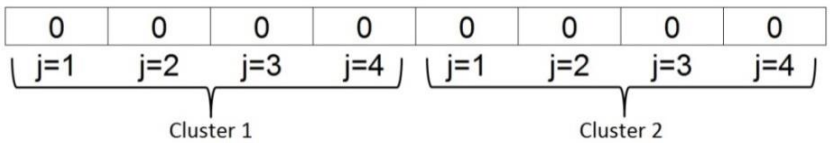
Setiap *Centroid* yang dibentuk terdiri dari 4 atribut, sama dengan jumlah atribut asli pada *dataset*. Pada contoh kasus diatas nilai pada masing-masing *centroid* diambil secara acak dari data didalam *dataset*. Misalkan pada *centroid* ke-1 nilai diambil dari *record* ke-1 sedangkan untuk *centroid* 2 nilai diambil dari record ke-10.

Pada algoritma HA-KPSO banyaknya partikel yang dibangkitkan disesuaikan dengan jumlah *Swarm size* yang didefinisikan sebelumnya. Jika *swarm size* diinputkan sebanyak 10 maka algoritma HA-KPSO akan membangkitkan populasi yang didalamnya berisi 10 buah partikel. contoh populasi partikel yang dibentuk pada algoritma HA-KPSO *clustering* dapat dilihat di Gambar 3.6.

Partikel 1	5,1	3,5	1,4	0,2	4,9	3,1	1,5	0,1
Partikel 2	6	2,2	4	1	5	3,3	1,4	0,2
Partikel 3	5,1	3,8	1,5	0,3	7,1	3	5,9	2,1
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…
Partikel 10	5,1	3,5	1,4	0,2	5	3,3	1,4	0,2

Gambar 3.6 Representasi *Swarm* Partikel

Partikel pada algoritma HA-KPSO selain menyimpan nilai *centroid* juga menyimpan nilai kecepatan. Tujuan dari fungsi kecepatan pada algoritma HA-KPSO adalah untuk merubah posisi suatu partikel menuju partikel dengan nilai *centroid* yang lebih optimal. Nilai inisialisasi awal dari kecepatan untuk setiap partikel pada algoritma HA-KPSO *clustering* adalah 0. Sebagai contoh Gambar 3.7 merupakan gambaran dari kecepatan partikel.



Gambar 3.7 Representasi Partikel dengan Nilai Kecepatan

Contoh nilai kecepatan partikel di Gambar 3.7 menunjukkan bahwa nilai 0 di kolom pertama merupakan kecepatan pada atribut ke-1, nilai 0 pada kolom kedua

merupakan kecepatan untuk atribut ke-2, nilai 0 pada kolom ketiga merupakan kecepatan untuk atribut ke-3 dan nilai 0 pada kolom keempat merupakan kecepatan untuk atribut ke-4. Kecepatan pada kolom pertama sampai kolom keempat merupakan kecepatan untuk *centroid* pada *cluster* 1. Hal ini juga berlaku pada nilai kecepatan di *centroid* lainnya.

2. Menghitung jarak antar data setiap Partikel

Langkah ini dilakukan setelah populasi terbentuk dan setiap partikel sudah memiliki nilai awal untuk *centroid* dan kecepatan. Selanjutnya adalah melakukan perhitungan jarak antara setiap partikel dengan setiap data didalam *dataset*. Fungsi untuk menghitung jarak yang digunakan algoritma HA-KPSO adalah *euclidean distance* dengan rumus yang dapat dilihat pada persamaan 2.1.

Setiap data yang telah dihitung nilai jaraknya dengan setiap *centroid* dimasukan ke dalam cluster yang memiliki nilai jarak terkecil. Proses ini terus dilakukan sampai seluruh data yang ada pada *dataset* telah memiliki kelompok *cluster*. Sebagai contoh hasil perhitungan jarak dan pemberian label *cluster* Pada algoritma HA-KPSO *clustering* dapat dilihat di Gambar 3.8.

id	centroid 1	centroid 2	cluster
1	0	2,828427125	1
2	1,414213562	2,236067977	1
3	2,414213562	2,628427125	1
…	…	…	…
…	…	…	…
150	4,123105626	3,16227766	2

Jarak data dengan centroid cluster ke-1 | Jarak data dengan centroid cluster ke-2

Gambar 3.8 Representasi perhitungan jarak dan Penentuan Cluster

Jika perhitungan jarak dan penentuan kelompok *cluster* untuk partikel ke-1 telah selesai, Maka perhitungan jarak akan dilanjutkan pada partikel ke-2. Proses perhitungan jarak akan terus diulang sampai kriteria berhenti terpenuhi, yaitu partikel terakhir dari populasi partikel yang dibangkitkan sebelumnya. Misalkan

populasi partikel yang dibangkitkan sebelumnya adalah 10 maka proses perhitungan jarak akan dilakukan sebanyak 10 kali.

3. Evaluasi Kinerja Partikel

Perhitungan kinerja dimulai dengan menentukan *cluster* setiap data dengan mencari jarak terkecil menggunakan rumus *euclidean distance* antara data dan nilai centroid setiap cluster dari partikel yang dilakukan pada tahapan sebelumnya. Proses ini merupakan proses untuk mengevaluasi setiap partikel dalam didalam populasi dengan menghitung nilai *fitness* setiap partikel dan mengevaluasinya sampai terpenuhi kriteria berhenti. Kriteria berhenti pada algoritma ini adalah banyaknya partikel didalam suatu populasi. Maka apabila jumlah partikel adalah 10 maka perhitungan nilai *fitness* akan dilakukan sebanyak 10 kali.

Suatu partikel dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran kinerjanya. Pada algoritma HA-KPSO *clustering* Sebuah partikel dianggap menjadi solusi optimal dengan melihat nilai *fitness* yang dihasilkan. Semakin kecil suatu nilai *fitness* maka semakin baik suatu partikel dijadikan sebuah solusi. Nilai *fitness* pada algoritma HA-KPSO *clustering* diperoleh dari fungsi validitas cluster. Fungsi validitas *cluster* yang digunakan pada algoritma ini adalah fungsi DBI, Maka persamaan yang digunakan untuk menghitung nilai fitness suatu partikel adalah fungsi DBI yang dapat dilihat pada persamaan 2.9.

Gambar 3.9 merupakan contoh populasi partikel *dataset* yang diperoleh dari inisialisasi partikel beserta hasil akhir perhitungan nilai *fitness* nya . Dalam satu populasi terdiri dari kumpulan partikel berjumlah 10 dengan jumlah populasi yang sudah ditentukan sebelumnya.

	Centroid 1				Centroid 2				Fitness
Partikel 1	5,1	3,5	1,4	0,2	4,9	3,1	1,5	0,1	0,48023
Partikel 2	6	2,2	4	1	5	3,3	1,4	0,2	0,49155
Partikel 3	5,1	3,8	1,5	0,3	7,1	3	5,9	2,1	0,48513
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…
Partikel 10	5,1	3,5	1,4	0,2	5	3,3	1,4	0,2	0,49275

Gambar 3.9 Contoh Representasi Nilai Fitness

4. Update Nilai awal *PBest* dan *GBest*

Langkah selanjutnya adalah melakukan inisialisasi dan pembaruan terhadap nilai *particle best* (*Pbest*) dan *global best*(*GBest*). Nilai *Pbest* dan *GBest* pada algoritma HA-KPSO digunakan untuk mencari perubahan kecepatan maupun posisi partikel. Solusi pencarian *centroid* terbaik pada algoritma HA-KPSO didapatkan dengan melakukan *update* tehadap nilai *Pbest* dan *Gbest* selama proses iterasi PSO berlangsung, sehingga menghasilkan partikel baru dengan nilai *centroid* yang lebih baik dari partikel sebelumnya.

Nilai *Pbest* didapatkan dengan membandingkan nilai *fitness* partikel pada iterasi saat ini dengan nilai partikel pada iterasi sebelumnya. *Pbest* pada iterasi pertama akan bernilai sama dengan posisi partikel saat pertama diinisialisasikan. Sedangkan untuk iterasi lebih dari satu, maka penentuan nilai *Pbest* dilakukan dengan membandingkan nilai *fitness* antar partikel. Partikel yang mempunyai nilai *fitness* terkecil yang akan terpilih sebagai nilai *Pbest* di iterasi saat ini.

5,1	3,5	1,4	0,2	4,9	3,1	1,5	0,1
6	2,2	4	1	5	3,3	1,4	0,2
5,1	3,8	1,5	0,3	7,1	3	5,9	2,1
…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
5,1	3,5	1,4	0,2	5	3,3	1,4	0,2

Gambar 3.10 Contoh Representasi Nilai *PBest* Pada Partikel

Global best didapatkan dari membandingkan semua partikel di seluruh iterasi PSO berdasarkan pada nilai *fitness*. Pada iterasi pertama, *Gbest* diperoleh dari partikel dalam satu populasi yang memiliki nilai fitness terkecil. Sebagai contoh misalkan pada populasi di Gambar 3.9, dimana partikel dengan nilai terkecil dan paling baik adalah partikel ke-1 sehingga partikel inilah yang dipilih sebagai *Gbest*. Sedangkan untuk iterasi lebih dari satu, *Gbest* akan melakukan perbandingan dengan nilai *fitnes* partikel yang didapat dari seluruh proses iterasi. Selama proses iterasi PSO berlangsung apabila ada partikel yang menghasilkan nilai fitness paling kecil, maka partikel tersebut akan menjadi *Gbest* algoritma

HA-KPSO. Nilai dari *Gbest* tidak akan berubah apabila selama iterasi tidak ditemukan lagi partikel dengan nilai *Gbest* yang lebih kecil.

5,1	3,5	1,4	0,2	4,9	3,1	1,5	0,1

Gambar 3.11 Contoh Representasi Nilai *GBest*

5. *Update* Posisi dan Kecepatan Partikel

Langkah selanjutnya adalah melakukan *update* kecepatan pada setiap partikel didalam populasi. Untuk mencari posisi partikel terbaik yang dapat menjadi kandidat solusi *centroid* dalam ruang pencarian, maka secara iteratif nilai kecepatan harus mengalami suatu perubahan. Nilai kecepatan pada partikel dapat dihitung dengan menggunakan persamaan 2.3. Nilai pada Parameter ω, C1,C2 , R1, dan R2 diinisialisasi terlebih dahulu pada awal proses algoritma HA-KPSO. Sedangkan nilai *Pbest* dan *Gbest* didapatkan dari inisialisasi di langkah sebelumnya. Perubahan kecepatan akan dilakukan secara berulang selama iterasi PSO berlangsung, sehingga nilai kecepatan dari seluruh partikel di iterasi sebelumnya akan berubah menjadi nilai kecepatan yang baru.

Setelah melalui proses untuk meng*update* nilai kecepatan di setiap partikel, selanjutnya akan dilakukan proses perubahan posisi untuk mencari posisi terbaik partikel. Untuk meng*update* posisi partikel dapat dilihat berdasarkan nilai posisi partikel pada iterasi sebelumnya dan nilai kecepatan pada iterasi saat ini. Kemudian dengan menggunakan persamaan 2.4, maka posisi partikel untuk iterasi saat ini dapat dihitung. Seperti halnya dengan perubahan nilai kecepatan, proses *update* posisi partikel juga dilakukan pada keseluruhan partikel didalam populasi.

6. Ulangi Langkah no 1 Selama *stopping criteria* Belum Terpenuhi

Stopping criteria yang digunakan pada proses pencarian *centroid* terbaik dari algoritma HA-KPSO adalah nilai maksimum dari iterasi PSO. Nilai maksimum iterasi yang digunakan pada penelitian ini adalah 10 iterasi. Disaat proses evaluasi partikel PSO sudah mencapai 10 kali iterasi, maka proses pengulangan algoritma akan dihentikan. Jika *stopping criteria* ini sudah tercapai maka proses akan

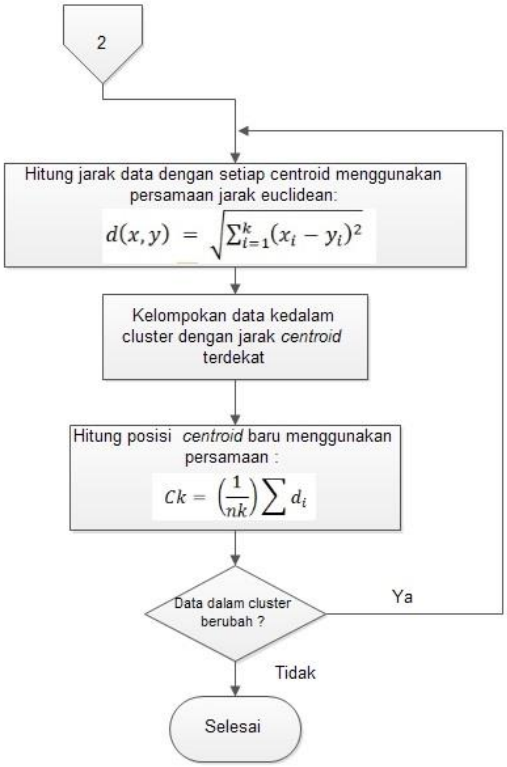
dilanjutkan ke langkah ke-7. Sebaliknya jika belum terpenuhi maka proses akan diulangi dari langkah no 1 dengan menggunakan posisi dan kecepatan partikel baru yang dihasilkan oleh iterasi saat ini.

7. Menentukan *Centroid*

Langkah terakhir dari pencarian *centroid* awal terbaik pada algoritma HA-KPSO adalah menentukan partikel mana yang menghasilkan solusi paling baik dari seluruh partikel didalam populasi. Partikel terbaik pada algoritma HA-KPSO diambil dari nilai *Gbest* yang didapakan setelah batas iterasi maksimum PSO terpenuhi. Setelah ditemukan nilai *Gbest* terbaik, maka solusi *centroid* terbaik dari keseluruhan populasi partikel juga didapatkan. Gambar 3.11 di halaman 35 Merupakan contoh dari nilai *centroid* terbaik.

3.2.3 Proses Pembentukan Cluster

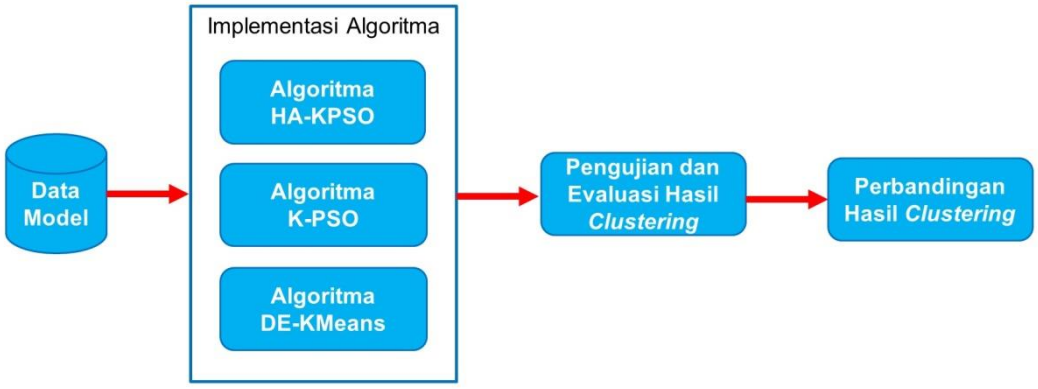
Langkah ketiga dari algoritma HA-KPSO adalah melakukan proses pembentukan *cluster* dengan cara melakukan pengelompokan setiap data ke dalam *cluster* menggunakan *centroid* yang dihasilkan pada proses sebelumnya sebagai *centroid* awal. Proses ini berbeda dengan algoritma K-Means yang membangkitkan *centroid* secara acak dari *dataset*. Pengelompokan data ke dalam *cluster* pada algoritma HA-KPSO menggunakan fungsi *Euclidean distance* sebagai parameter pembanding. Setiap data akan dihitung nilai jaraknya dengan setiap *centroid*. Kemudian data akan dikelompokan ke dalam *cluster* yang *centroid*nya memiliki jarak terdekat dengan data. Proses pengelompokan data ke dalam *cluster* akan terus dilakukan sampai kondisi data yang ada didalam *cluster* tidak mengalami perubahan dari hasil *cluster* sebelumnya. Hasil akhir proses pembentukan *cluster* pada algoritma HA-KPSO adalah kumpulan cluster yang telah berisi data yang sesuai dengan karakteristik dari *cluster* tersebut. Berikut adalah flowchart proses pembentukan *cluster* algoritma HA-KPSO.



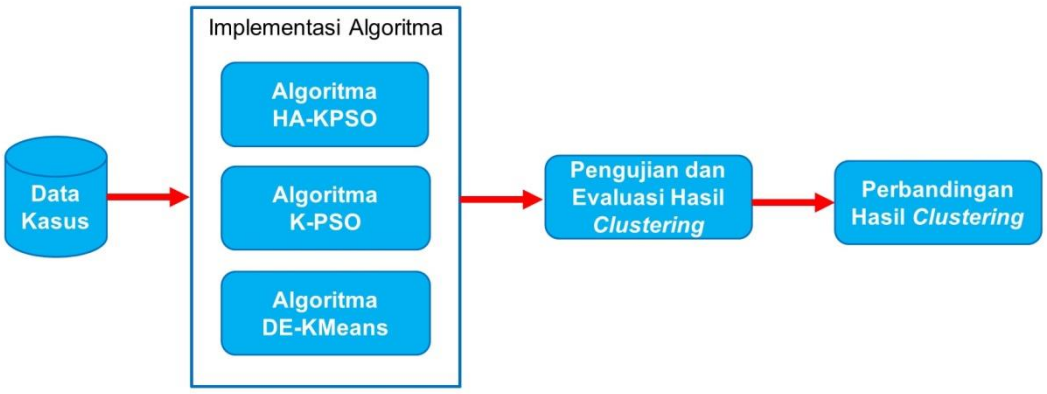
Gambar 3.12 Flowchart Proses Pembentukan *Cluster* Algoritma HA-KPSO

3.3 Teknik Penelitian

Pada penelitian ini, Teknik penelitian dibagi mejadi dua tahap yaitu pengujian terhadap data model dan pengujian terhadap data kasus seperti yang ditampilkan di Gambar 3.13 dan Gambar 3.14.



Gambar 3.13 Alur Pengujian Terhadap Data Model



Gambar 3.14 Alur Pengujian Terhadap Data Kasus

Pengujian terhadap data model dilakukan untuk mengetahui karakteristik dari algoritma HA-KPSO terhadap data secara menyeluruh. Kemudian pengujian dilakukan terhadap data kasus untuk dapat mengetahui kualitas hasil *clustering* algoritma HA-KPSO apabila diterapkan pada kasus segmentasi pasar film. Data kasus yang digunakan pada penelitian ini adalah data film yang diambil dari IMDB. Penjelasan lebih lengkap mengenai data penelitian dan alur pengujian yang digunakan dapat dibaca di Subbab 3.3.1 dan Subbab 3.3.2.

3.3.1 Data Penelitian

Untuk melakukan pengujian terhadap algoritma HA-KPSO, diperlukan suatu data uji agar dapat diketahui bagaimana hasil dan kinerja dari algoritma tersebut. Data yang digunakan pada penelitian ini dibagi menjadi dua jenis yaitu data model dan data kasus.

3.3.1.1 Data Model

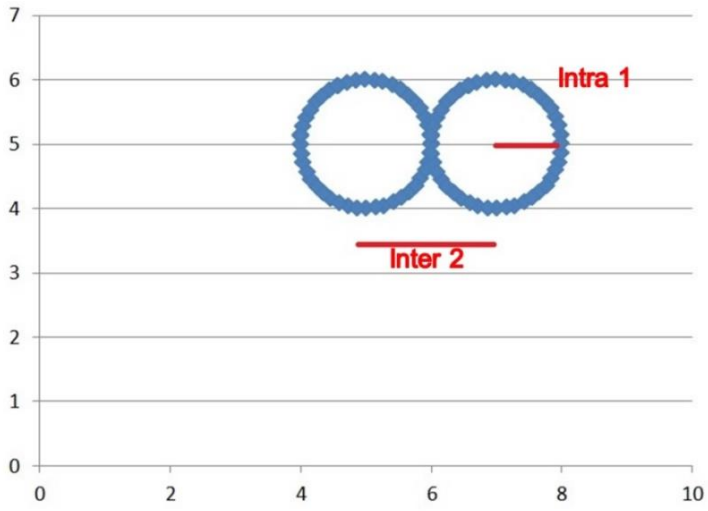
Data Model merupakan *dataset* yang digunakan untuk melihat karakteristik dari algoritma HA-KPSO terhadap data secara menyeluruh. Data model yang digunakan pada pengujian penelitian ini dibagi menjadi dua kategori yaitu data model buatan dan data model UCI. Data model buatan merupakan *dataset* yang dibuat sendiri oleh peneliti dengan sudah mengetahui karakteristik dari bentuk, jumlah *cluster* dan jarak antar *cluster* nya terlebih dahulu. Sedangkan data model

UCI merupakan kumpulan *dataset* yang umum digunakan oleh para peneliti di bidang *clustering* untuk menguji berbagai algoritma dalam menyelesaikan kasus *clustering*.

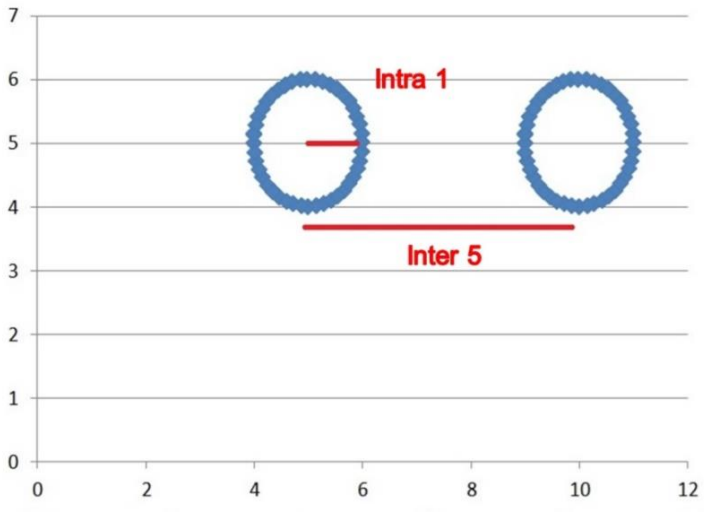
1. *Dataset* Buatan

Dataset buatan adalah *dataset* yang dimanipulasi sebaran data, jumlah data, jumlah atribut serta jumlah *cluster*-nya. Data model buatan digunakan untuk mengetahui perilaku algoritma terhadap data. Pada penelitian ini jumlah atribut yang digunakan pada *dataset* dimulai dari 2 atribut hingga 13 atribut, sementara bentuk sebaran data dibuat dalam bentuk 2 cluster dan 3 *cluster*. Variasi jumlah data pada *dataset* 2 *cluster* dimulai dari 24, 48, 96, 192, 384 dan 768 data. Sementara variasi jumlah data pada *dataset* 3 *cluster* dimulai dari 36, 72, 144, 288, 576 dan 1152 data. Pemilihan jumlah atribut dan jumlah data yang digunakan pada *dataset* buatan didasarkan pada jumlah atribut dan jumlah data dari data model UCI yang digunakan, Sehingga dapat memudahkan dalam membandingkan hasil pengujian algoritma HA-KPSO.

Manipulasi sebaran data untuk *dataset* buatan dilakukan dengan cara menjauhkan jarak antar *cluster* seperti yang dapat dilihat di Gambar 3.15 dan Gambar 3.16.



Gambar 3.15 Contoh Sebaran Dataset Buatan Intra 1 dan Inter 2



Gambar 3.16 Contoh Sebaran Dataset Buatan Intra 1 dan Inter 5

Gambar 3.15 dan Gambar 3.16 memperlihatkan contoh *dataset* buatan dengan jumlah 2 atribut yang memiliki 48 data pada masing-masing *cluster*. Kedua gambar tersebut memperlihatkan perbedaan sebaran data dimana *dataset* buatan intra 1 dan inter 2 memiliki jarak *cluster* yang berdekatan sementara *dataset* buatan intra 1 dan inter 5 memiliki jarak yang lebih jauh.

2. *Dataset* UCI *Machine Learning*

Dataset UCI *machine learning* merupakan kumpulan *dataset* yang umum digunakan oleh para peneliti untuk menguji berbagai algoritma dalam menyelesaikan kasus *clustering* data. Dengan menggunakan standar *dataset* yang sama, maka peneliti yang berbeda dapat membandingkan hasil dari algoritma yang mereka buat baik secara visual maupun kuantitatif. Dataset UCI *machine learning* diperoleh dari situs https://archive.ics.uci.edu/ yang disediakan oleh Nationan Science Foundation. Situs terebut menyediakan berbagai macam *dataset* yang khusus digunakan untuk proses *data mining*. Dari sekian banyak dataset uji yang tersedia, penelitian ini hanya menggunakan tiga buat *dataset* yang sudah dipilih bedasarkan karakteristik jumlah data dan jumlah atribut. Tiga *dataset* uci yang dipilih pada penelitian ini yaitu dataset Iris, Diabetes dan Wine. Karakteristik dari ketiga dataset tersebut dapat dilihat di Tabel 3.1.

Tabel 3.1 Karakteristik Dataset UCI

Dataset	Jumlah Data	Jumlah Atribut	Nilai Atribut yang Hilang	Keterangan
Iris	150	4	Tidak	Pengelompokkan bunga Iris berdasarkan panjang dan lebar kelopak dan daun bunga
Diabetes	768	8	Ya	Kemungkinan seseorang terkena penyakit diabetes berdasarkan kriteria tertentu seperti indeks masa tubuh, usia, dll.
Wine	178	13	Tidak	Penentuan tipe wine (anggur) berdasarkan kriteria tertentu seperti warna, kandungan kimia, dll.

A. Dataset Iris

Dataset iris merupakan dataset yang paling sering dijumpai pada literatur yang membahas penelitian di bidang *clustering* dan pengenalan pola. Dataset ini ini terdiri dari 150 *record* dan 4 atribut ditambah dengan sebuah atribut kelas. Penelitian ini hanya menggunakan 4 atribut dari *dataset* Iris yaitu *sepal lenght*, *sepal width*, *petal length*, *petal width*. Atribut kelas iris tidak digunakan karena pada clustering tidak memerlukan atribut kelas dalam pembentukan *cluster*. Ke-empat atribut yang digunakan merupakan representasi dari panjang dan lembar kelopak serta mahkota bunga dengan tipe data numerik.. Contoh dari dataset Iris dapat dilihat di Gambar 3.17

Sepal Length	Sepal Width	Petal length	Petal Width
5,1	3,5	1,4	0,2
4,9	3	1,4	0,2
4,7	3,2	1,3	0,2
4,6	3,1	1,5	0,2
5	3,6	1,4	0,2
5,4	3,9	1,7	0,4
4,6	3,4	1,4	0,3
5	3,4	1,5	0,2
4,4	2,9	1,4	0,2
4,9	3,1	1,5	0,1
5,4	3,7	1,5	0,2
4,8	3,4	1,6	0,2
4,8	3	1,4	0,1

Gambar 3.17 Dataset Iris

B. Dataset Diabetes

Dataset Diabetes merupakan dataset yang sering dijumpai pada literatur yang membahas penelitian di bidang *clustering* dan pengenalan pola. Dataset diabetes merepresentasikan pasien wanita yang memiliki kemungkinan untuk mengidap penyakit diabetes. *Dataset* terdiri dari 768 *record* dan 8 atribut ditambah sebuah *atribut* kelas. Penelitian ini hanya menggunakan 8 atribut dari *dataset* diabetes yaitu *pregnancies, PG Concetration, Diastolik BP, Tri fold Thick, Serum Ins, BMI, DP Fuction*, dan *Age*. Atribut kelas Diabetes tidak digunakan karena pada *clustering* tidak memerlukan atribut kelas dalam pembentukan cluster. Ke-delapan atribut yang digunakan merupakan representasi dari gejala penyakit diabetes pada wanita dengan masing-masing atribut memiliki tipe data *numerik*. Contoh dataset diabetes dapat dilihat di Gambar 3.18.

Pregnancies	PG Concentration	Diastolic BP	Tri Fold Thick	Serum Ins	BMI	DP Function	Age
6	148	72	35	0	33,6	0,627	50
1	85	66	29	0	26,6	0,351	31
8	183	64	0	0	23,3	0,672	32
1	89	66	23	94	28,1	0,167	21
0	137	40	35	168	43,1	2,288	33
5	116	74	0	0	25,6	0,201	30
3	78	50	32	88	31	0,248	26
10	115	0	0	0	35,3	0,134	29
2	197	70	45	543	30,5	0,158	53
8	125	96	0	0	0	0,232	54
4	110	92	0	0	37,6	0,191	30
10	168	74	0	0	38	0,537	34
10	139	80	0	0	27,1	1,441	57
1	189	60	23	846	30,1	0,398	59
5	166	72	19	175	25,8	0,587	51
7	100	0	0	0	30	0,484	32
0	118	84	47	230	45,8	0,551	31
7	107	74	0	0	29,6	0,254	31
1	103	30	38	83	43,3	0,183	33

Gambar 3.18 Dataset Diabetes

C. Dataset Wine

Dataset wine mungkin merupakan *dataset* yang sering dijumpai pada literatur yang membahas penelitian di bidang *clustering* dan pengenalan pola. Dataset wine berisi hasil analisis reaksi kimia pada anggur yang tumbuh pada wilayah yang sama di Italia namun berasal dari tiga kultivar yang berbeda. *Dataset* terdiri dari

178 *record* dan 13 atribut ditambah dengan sebuah atribut kelas. Penelitian ini hanya menggunakan 13 atribut dari *dataset* wine yaitu *alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, Nonflavanoid phenols, proanthocyanins, color intensity, hue, dan proline*. Atribut kelas wine tidak digunakan karena pada *clustering* tidak memerlukan atribut kelas dalam pembentukan cluster. Ke-tigabelas atribut yang digunakan merupakan representasi dari reaksi kimia yang terdapat pada anggur dengan setiap atribut memiliki tipe data *numerik*. Contoh dari *dataset* wine dapat dilihat di Gambar 3.19

Alcohol	Malic acid	Ash	Alcalinity of ash	Magnesium	Total phenols	Flavanoids	Nonflavanoid phenols	Proanthocyanins	Color intensity	Hue	OD280/OD315 of diluted wines	Proline
14,23	1,71	2,43	15,6	127	2,8	3,06	0,28	2,29	5,64	1,04	3,92	1065
13,2	1,78	2,14	11,2	100	2,65	2,76	0,26	1,28	4,38	1,05	3,4	1050
13,16	2,36	2,67	18,6	101	2,8	3,24	0,3	2,81	5,68	1,03	3,17	1185
14,37	1,95	2,5	16,8	113	3,85	3,49	0,24	2,18	7,8	0,86	3,45	1480
13,24	2,59	2,87	21	118	2,8	2,69	0,39	1,82	4,32	1,04	2,93	735
14,2	1,76	2,45	15,2	112	3,27	3,39	0,34	1,97	6,75	1,05	2,85	1450
14,39	1,87	2,45	14,6	96	2,5	2,52	0,3	1,98	5,25	1,02	3,58	1290
14,06	2,15	2,61	17,6	121	2,6	2,51	0,31	1,25	5,05	1,06	3,58	1295
14,83	1,64	2,17	14	97	2,8	2,98	0,29	1,98	5,2	1,08	2,85	1045
13,86	1,35	2,27	16	98	2,98	3,15	0,22	1,85	7,22	1,01	3,55	1045
14,1	2,16	2,3	18	105	2,95	3,32	0,22	2,38	5,75	1,25	3,17	1510
14,12	1,48	2,32	16,8	95	2,2	2,43	0,26	1,57	5	1,17	2,82	1280
13,75	1,73	2,41	16	89	2,6	2,76	0,29	1,81	5,6	1,15	2,9	1320
14,75	1,73	2,39	11,4	91	3,1	3,69	0,43	2,81	5,4	1,25	2,73	1150
13,63	1,81	2,7	17,2	112	2,85	2,91	0,3	1,46	7,3	1,28	2,88	1310
14,3	1,92	2,72	20	120	2,8	3,14	0,33	1,97	6,2	1,07	2,65	1280
13,83	1,57	2,62	20	115	2,95	3,4	0,4	1,72	6,6	1,13	2,57	1130
14,19	1,59	2,48	16,5	108	3,3	3,93	0,32	1,86	8,7	1,23	2,82	1680
13,64	3,1	2,56	15,2	116	2,7	3,03	0,17	1,66	5,1	0,96	3,36	845

Gambar 3.19 Dataset Wine

3.3.1.2 Data Kasus

Penelitian ini berfokus pada *clustering* untuk menghasilkan *cluster* yang memiliki nilai DBI terbaik pada kasus segmentasi pasar film. Supaya pembagian segmentasi pasar dapat dilakukan dengan tepat, maka diperlukan sebuah sumber data yang menyimpan dan menyediakan berbagai data mengenai karakteristik suatu film. Salah satu sumber yang dapat dipakai untuk mengetahui informasi tentang film adalah *International Movie Database* (IMDB). IMDB merupakan sebuah *website* yang menyediakan dan menyimpan data mengenai film. Berbagai data film dapat ditemukan disini seperti data genre, tahun tayang, budget, rating dan berbagai data lainnya. IMDB sering digunakan oleh pengguna internet untuk mencari dan membagikan informasi mengenai masalah perfilman. Di website IMDB pengguna internet dapat memberikan review, vote, atau kritik pada film dan hasil review tersebut kemudian disimpan pada database sistem. Oleh sebab itu

IMDB memiliki sumber data yang lengkap untuk menemukan informasi dari berbagai film yang pernah diproduksi.Bedasarkan alasan tersebut, maka penelitian ini akan menggunakan IMDB sebagai data untuk kasus segmentasi pasar film.

Pada penelitian ini, atribut pada *dataset* IMDB yang digunakan untuk mengetahui segmentasi pasar film adalah atribut genre, musim tayang, rating konten, *budget*, pendapatan kotor, pendapatan bersih, jumlah vote, jumlah *review*, jumlah kritik, jumlah like aktor, jumlah like actor, jumlah like director dan skor Film. Akan tetapi masih belum diketahui bagaimana bentuk karakteristik, jumlah kelompok pasar yang terbentuk, dan anggota dari masing-masing kelompok pada *dataset* IMDB. Untuk menyelesaikan masalah tersebut akan digunakan teknik clustering menggunakan algoritma HA-KPSO, sehingga data film dapat dikelompokan berdasarkan kemiripan antar datanya. Proses *clustering* akan dilakukan terhadap 500 data film yang tayang dari tahun 2014-2016. Contoh dari *dataset* IMDB dapat dilihat di Gambar 3.20.

ID	genre	Musim	Rating Konten	budget	Pendapatan Kotor	Pendapatan Bersih	Jumlah Vote	Jumlah Review	Jumlah Kritik	Total Like Director	Total Like Aktor	Score IMDB
1	Sci-Fi	Panas	PG-13	237000000	760505847	523505847	886204	3054	723	23000	4834	7.9
2	Fantasy	Panas	PG-13	300000000	309404152	9404152	471220	1238	302	19563	48350	7.1
3	Thriller	Semi	PG-13	245000000	400074175	155074175	275868	994	602	203	11700	6.8
4	Thriller	Gugur	PG-13	250000000	448130642	198130642	1144337	2701	813	22000	106759	8.5
5	Sci-Fi	Dingin	PG-13	263700000	73058679	-190641321	212204	738	462	9751	1873	6.6
6	Sci-Fi	Panas	PG-13	258000000	336530303	78530303	383056	1902	392	204	46055	6.2
8	Animation	Dingin	PG	260000000	200807262	-59192738	294810	387	324	15	2036	7.8
9	Sci-Fi	Semis	PG-13	250000000	458991599	208991599	462669	1117	635	20000	92000	7.5
10	Fantasy	Gugur	PG	250000000	301956980	51956980	321795	973	375	11282	58753	7.5
11	Sci-Fi	Semi	PG-13	250000000	330249062	80249062	371639	3018	673	8500	24450	6.9
12	Sci-Fi	Panas	PG-13	209000000	200069408	-8930592	240396	2367	434	11430	29991	6.1
13	Thriller	Gugur	PG-13	200000000	168368427	-31631573	330784	1243	403	395	2023	6.7
14	Fantasy	Panas	PG-13	225000000	423032628	198032628	522040	1832	313	19563	48486	7.3
15	Western	Semi	PG-13	215000000	89289910	-125710090	181792	711	450	19563	45757	6.5
16	Sci-Fi	Semi	PG-13	225000000	291021565	66021565	548573	2536	733	8500	20495	7.2
17	Fantasy	Panas	PG	225000000	141614023	-83385977	149922	438	258	800	22697	6.6
18	Sci-Fi	Panas	PG-13	220000000	623279547	403279547	995415	1722	703	20000	87697	8.1
19	Fantasy	Panas	PG-13	250000000	241063875	-8936125	370704	484	448	252	54083	6.7
20	Comedy	Gugur	PG-13	225000000	179020854	-45979146	268154	451	451	2188	12572	6.8

Gambar 3.20 *Dataset* film IMDB

Sebelum menjalankan proses *clustering* menggunakan algoritma HA-KPSO, perlu dilakukan proses transformasi data terlebih dahulu. Atribut yang mempunyai tipe data diluar numerik seperti atribut genre, rating konten, dan musim tayang harus diubah ke bentuk tipe data numerik terlebih dahulu untuk dapat digunakan dalam proses clustering. Proses transformasi menggunakan langkah-langkah sebagai berikut:

1. Pada atribut genre, rating konten dan musim terlebih dahulu dilakukan pengurutan berdasarkan frekuensi kemunculan data yang paling banyak muncul hingga yang paling sedikit muncul
2. Data yang memiliki frekuensi kemunculan paling sedikit diberi inisial dengan angka 1, sedangkan data dengan frekuensi kemunculan lebih banyak dari data pertama akan diberi inisial angka 2 dan begitu seterusnya hingga genre dengan frekuensi kemunculan paling banyak.

Hasil tranformasi dari atribut genre dapat dilihat di Tabel 3.2

Tabel 3.2 Hasil Transformasi Atribut Genre

Genre	Frekuensi	Inisial
Western	6	1
Horror	11	2
History	22	3
Sport	26	4
Mystery	26	5
Romance	40	6
Comedy	45	7
Fantasy	78	8
Animation	81	9
Triller	89	10
Sci-Fi	95	11

Hasil tranformasi dari atribut rating konten dapat dilihat di Tabel 3.3.

Tabel 3.3 Hasil Transformasi Atribut Rating Konten

Rating Konten	Frekuensi	Inisial
G	22	1
R	78	2
PG	128	3
PG-13	272	4

Hasil tranformasi dari atribut musim tayang dapat dilihat di Tabel 3.4.

Tabel 3.4 Hasil Transformasi Atribut Musim Tayang

Rating Konten	Inisial
Dingin (Januari - Maret)	1
Semi (April - Juni)	2
Panas (Juli - September)	3
Gugur (Oktober-Desember)	4

Proses transformasi selanjutnya adalah merubah nilai pada atribut budget dan pendapatan bersih. Proses transformasi dilakukan dengan cara mengelompokan setiap data ke dalam suatu rentang nilai. Setiap data pada atribut kemudian akan diberi nilai inisial sesuai kelompok rentang data nya. Hasil transformasi dari atribut budget dilihat di Tabel 3.5

Tabel 3.5 Hasil Transformasi Atribut Budget

Budget	Inisial
$0 - $50000000	1
$5000000 - $100000000	2
$100000000 - $150000000	3
$150000000 - $200000000	4
$200000000 - $250000000	5
> $250000000	6

Hasil transformasi dari atribut pendapatan bersih dilihat di Tabel 3.6

Tabel 3.6 Hasil Transformasi Atribut Pendapatan Bersih

Budget	Inisial
< $0	1
$ 0 - $100000000	2
$100000000 - $200000000	3
$200000000 - $300000000	4
> $300000000	5

Setelah proses transformasi data, maka *dataset* IMDB siap digunakan untuk pengujian data kasus. *Dataset* ini yang selanjutnya akan digunakan sebagai data masukan untuk pengujian data kasus di BAB 4. Gambar 3.21 merupakan contoh *dataset* imdb yang siap digunakan pada pengujian data kasus.

ID	Genre	Musim	Rating Konten	budget	Pendapatan Kotor	Pendapatan Bersih	Jumlah Vote	Jumlah Review	Jumlah Kritik	Total Like Direktor	Total Like Aktor	Skor IMDB
1	11	3	4	5	760505847	8	886204	3054	723	23000	4834	7.9
2	8	3	4	7	309404152	2	471220	1238	302	19563	48350	7.1
3	10	2	4	5	400074175	1	275868	994	602	203	11700	6.8
4	10	4	4	6	448130642	5	1144337	2701	813	22000	106759	8.5
5	11	1	4	6	73058679	1	212204	738	462	9751	1873	6.6
6	11	3	4	6	336530303	3	383056	1902	392	204	46055	6.2
8	9	1	3	6	200807262	1	294810	387	324	15	2036	7.8
9	11	2	4	6	458991599	6	462669	1117	635	20000	92000	7.5
10	8	4	3	6	301956980	3	321795	973	375	11282	58753	7.5
11	11	2	4	6	330249062	3	371639	3018	673	8500	24450	6.9
12	11	3	4	5	200069408	1	240396	2367	434	11430	29991	6.1
13	10	4	4	4	168368427	1	330784	1243	403	395	2023	6.7
14	8	3	4	5	423032628	5	522040	1832	313	19563	48486	7.3
15	1	2	4	5	89289910	1	181792	711	450	19563	45757	6.5
16	11	2	4	5	291021565	3	548573	2536	733	8500	20495	7.2
17	8	3	3	5	141614023	1	149922	438	258	800	22697	6.6
18	11	3	4	5	623279547	8	995415	1722	703	20000	87697	8.1
19	8	3	4	6	241063875	1	370704	484	448	252	54083	6.7
20	7	4	4	5	179020854	1	268154	451	451	2188	12572	6.8

Gambar 3.21 Dataset film IMBD Untuk Pengujian Kasus

3.3.2 Tahapan Pengujian

Tahapan Pengujian pada penelitian ini terdiri dari dua tahapan yaitu pengujian terhadap data model dan pengujian terhadap data kasus.

3.3.2.1 Pengujian Data Model

Pengujian data model bertujuan untuk untuk melihat karakteristik dan pola perilaku dari algoritma HA-KPSO terhadap data, sehingga nantinya dapat diimplementasikan untuk kasus yang tepat sesuai karakter dari algoritma HA-KPSO. Pengujian data model dilakukan dengan cara mengimplementasikan algoritma *clustering* yang diusulkan yaitu algoritma HA-KPSO dan beberapa algoritma pembanding terhadap data model yang terdiri dari *dataset* buatan dan *dataset* UCI. Pengujian dimulai dengan melakukan inisialisasi nilai terhadap parameter PSO yang digunakan oleh algoritma HA-KPSO. Nilai setiap Parameter PSO yang digunakan pada pengujian data model dapat dilihat di Tabel 3.7

Tabel 3.7 Nilai parameter PSO

No	Parameter PSO	Inisialisasi Nilai
1	*Swarm Size*	10
2	ω	0,72
3	C1	1,5
4	C2	1,5
5	R1	Nilai acak antara 0-1
6	R2	Nilai acak antara 0-1
7	*Stopping criteria*	10

Selanjutnya seluruh algoritma akan diuji keseluruh data model yang sudah disiapkan untuk mendapatkan hasil *clustering*. Hasil *clustering* yang terbentuk akan dievaluasi menggunakan metode *Davies-Bouldin Index* (DBI). Pada metode DBI, Semakin kecil nilai DBI menunjukkan semakin baik hasil *clustering* tersebut, sebaliknya semakin besar nilai DBI menunjukkan semakin buruk hasil *clustering* yang dihasilkan. Hal ini dilakukan untuk mengetahui bagaimana kinerja algoritma dari kualitas cluster yang dihasilkan berdasarkan teknik *clustering* yang berbeda, Sehingga dapat diketahui kecocokan antara algoritma, data, dan teknik evaluasi *clustering* yang digunakan.

Selain melihat kinerja algoritma dari kualitas cluster yang dihasilkan, Algoritma juga akan diuji untuk mengetahui *cost* yang harus dikeluarkan dalam proses *clustering*. Jumlah iterasi dapat menunjukkan *cost* yang diperlukan dalam proses *clustering* algoritma, sehingga semakin sedikit jumlah iterasi artinya semakin cepat proses pembentukan *clsuter* dilakukan.Sebaliknya semakin banyak jumlah iterasi artinya semakin lama pula proses pembentukan cluster dilakukan.

Pada penelitian ini pengujian terhadap data model akan dilakukan sebanyak 10 kali percobaan pada setiap dataset yang diuji. Hasil yang akan ditampilkan merupakan rata-rata dari DBI dan jumlah iterasi dari 10 kali percobaan. Selanjutnya untuk mengetahui tingkat keakuratan algoritma dari 10 kali percobaan, Maka akan dihitung nilai standar deviasi nya. Hasil dari pengujian dan analisis terhadao data model akan dibahas lebih lanjut di BAB 4.

3.3.2.2 Pengujian Data Kasus

Pengujian data kasus bertujuan untuk untuk menggabarkan hasil implementasi algoritma terhadap kasus dunia nyata, sehingga nantinya algoritma HA-KPSO dapat memberikan informasi yang tepat dalam pengambilan keputusan strategis. Pengujian kasus dilakukan dengan cara mengimplementasikan algoritma *clustering* yang diusulkan yaitu algoritma HA-KPSO dan beberapa algoritma pembanding terhadap data kasus yang sudah disiapkan yaitu data film imdb. Pengujian dimulai dengan melakukan inisialisasi nilai terhadap parameter PSO yang digunakan oleh algoritma HA-KPSO. Nilai setiap Parameter PSO untuk pengujian data kasus akan menggunakan nilai yang sama dengan pengujian data model yang disajikan di Tabel 3.7.

Selanjutnya seluruh algoritma akan diuji pada data kasus yang sudah disiapkan yaitu *dataset* IMDB untuk mendapatkan hasil *clustering*. Hasil clustering yang terbentuk dari setiap algoritma akan dievaluasi menggunakan menggunakan metode *Davies-Bouldin Index* (DBI). Pada metode DBI, Semakin kecil nilai DBI menunjukkan semakin baik hasil *clustering* tersebut, sebaliknya semakin besar nilai DBI menunjukkan semakin buruk hasil *clustering* yang dihasilkan. Hal ini dilakukan untuk mengetahui bagaimana kinerja algoritma dari

kualitas cluster yang dihasilkan berdasarkan teknik *clustering* yang berbeda, Sehingga dapat diketahui kecocokan antara algoritma, data, dan teknik evaluasi *clustering* yang digunakan.

Selain melihat kinerja algoritma dari kualitas cluster yang dihasilkan, Algoritma juga akan diuji untuk mengetahui *cost* yang harus dikeluarkan dalam proses *clustering*. Jumlah iterasi dapat menunjukkan *cost* yang diperlukan dalam proses *clustering* algoritma, sehingga semakin sedikit jumlah iterasi artinya semakin cepat proses pembentukan *clsuter* dilakukan.Sebaliknya semakin banyak jumlah iterasi artinya semakin lama pula proses pembentukan cluster dilakukan. Hasil dari pengujian dan analisis terhadap data model akan dibahas lebih lanjut di BAB 4.