DATA MINING

11 – DBSCAN

Oleh: Leny Tritanto N., M.Kom.

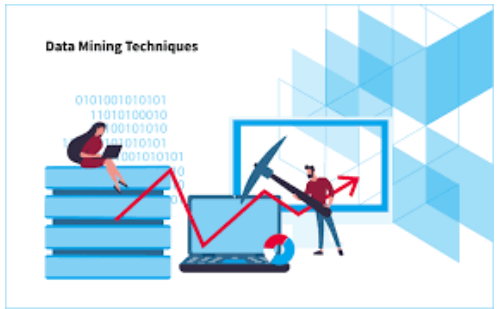
Sumber: (DBSCAN Clustering, algotech.netlify.app), (Handsout Data Mining, Lis Utari)

UNBIN

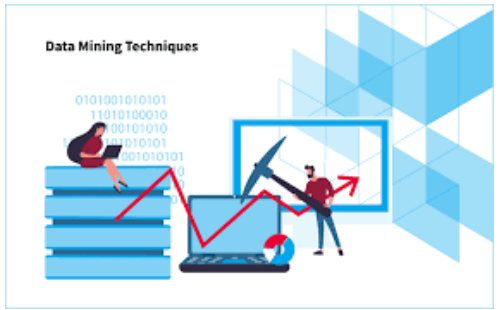
Algoritma Supervised dan UnsupervisedLearning:

Supervised Learning	Unsupervised Learning
▪ Linear Regression	✓ K-Means
✓ Decision Tree and Random Forest	✓ Hierarchical Clustering
✓ Naive Bayes Classifier	➡ DBSCAN
✓ Nearest Neighbour Classifier (KNN)	▪ Association Rule
✓ Artificial Neural Network	▪ Apriori Algorithm
✓ Support Vector Machine (SVM)	

UNBIN



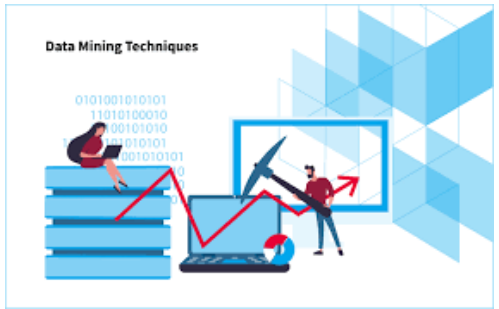
Apa dan bagaimana menerapkan Algoritma Clustering dengan DBSCAN?



Introduction to DBSCAN Clustering

DBSCAN atau *Density Based Spatial Clustering of Application with Noise* adalah metode clustering yang berbasis kepadatan *(density-based)* dari posisi amatan data dengan prinsip mengelompokkan data yang relatif berdekatan. DBSCAN sering diterapkan pada data yang banyak mengandung noise, hal ini dikarenakan DBSCAN tidak akan memasukkan data yang dianggap noise kedalam cluster manapun.

UNBIN



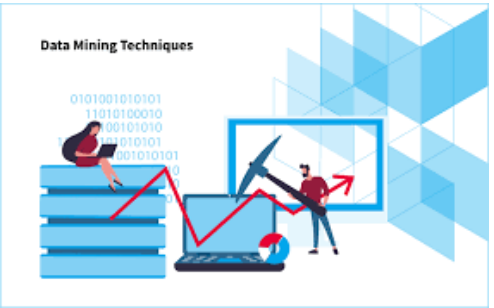
Introduction to DBSCAN Clustering

- DBSCAN memiliki keunggulan berupa performa untuk menangkap cluster yang memiliki beragam bentuk
- Metode DBSCAN kurang cocok digunakan pada data dengan tingkat kerapatan beragam
- DBSCAN juga kurang cocok digunakan pada data dengan dimensi yang terlalu besar

Catatan:

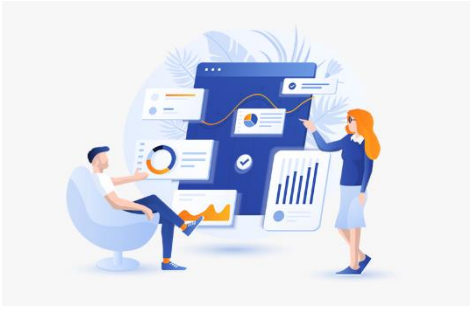
- Untuk menghasilkan cluster yang dibutuhkan, sebaiknya metode ini dilakukan oleh pengguna yang sudah benar-benar memahami seluk beluk metode ini dengan baik, hal ini perlu dilakukan untuk menghindari terjadinya error pada data yang diolah.
- Selama menggunakan algoritma ini data yang diolah bisa saja mengalami kerusakan sehingga sangat penting untuk memahami metode ini dengan benar sebelum menggunakan agar proses *clustering* berjalan sesuai harapan.

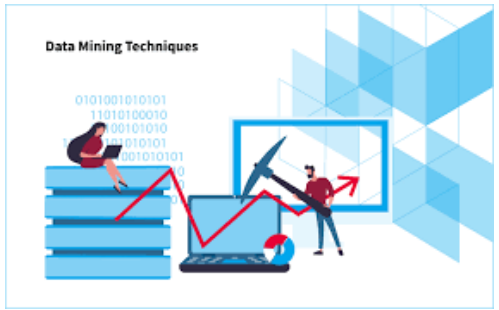
UNBIN



Introduction to DBSCAN Clustering

- Clustering atau kelompok data yang baik adalah cluster yang memiliki kemiripan yang besar antar anggota clusternya dan memiliki perbedaan yang signifikan dengan anggota cluster yang berbeda
- Clustering dapat diterapkan untuk mengolah data dalam berbagai bidang seperti segmentasi pasar, data spasial serta cluster profiling.
- Clustering dikelompokan menjadi beberapa metode, yaitu: partisi, hirarki, density-based, grid-based

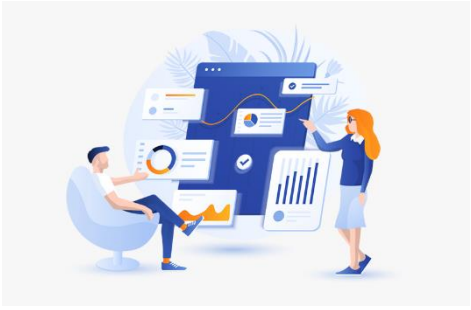


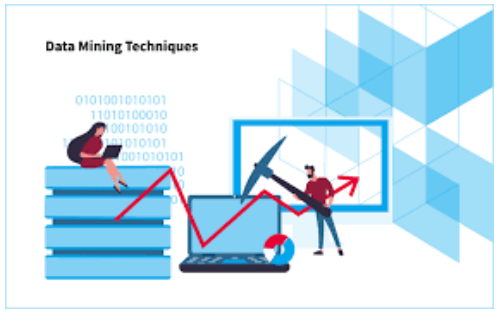


Introduction to DBSCAN Clustering (Terminologi)

- DBSCAN sering digunakan pada data yang mengandung banyak *noise* atau gangguan
- DBSCAN cenderung memisahkan data yang mengandung *noise* agar tidak bercampur cluster lain
- DBSCAN memerlukan dua parameter input sebelum proses clustering, yaitu epsilon (eps) dan minimum points (minPts)
- Epsilon merupakan jarak maksimal antara dua data dalam satu cluster, dan minimum points adalah banyaknya data minimal dalam jarak epsilon agar terbentuk satu cluster.
- Selain epsilon dan minPts, terdapat beberapa terminologi lain yang berlaku dalam proses clustering dengan metode DBSCAN yaitu:

  - ***directly density-reachable*** : Observasi q berhubungan langsung dengan p, jika p adalah core point dan q merupakan tetangga dari p dalam jangkauan epsilon.
  - ***density-reachable*** : Observasi q dan x dalam satu cluster namun x bukan tetangga dari q dalam jangkauan epsilon.
  - ***core point*** : Core point merupakan observasi yang memiliki jumlah tetangga lebih dari sama dengan dari MinPts pada jangkauan Eps.
  - ***border point*** : Border point memiliki tetangga lebih sedikit dari Minpts namun ia merupakan tetangga dari core point.
  - ***outlier/noise point*** : Observasi yang bukan border points atau core points.





DBSCAN Procedure

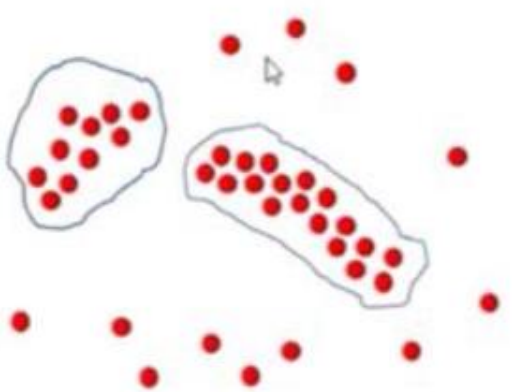
Dalam proses pembuatan cluster menggunakan DBSCAN sebuah data akan dikelompokkan dengan tetangganya. Sepasang amatan dikatakan bertetangga apabila jarak antara dua amatan tersebut kurang dari sama dengan nilai epsilon. Secara sederhana cara kerja DBSCAN adalah sebagai berikut :

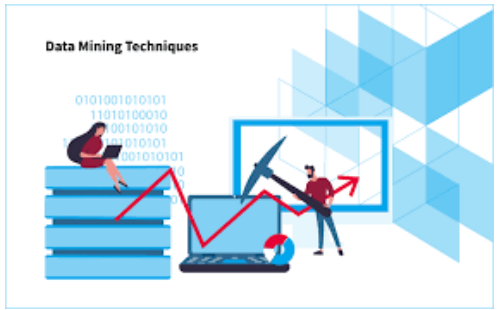
1. Tentukan nilai minPts dan epsilon (eps) yang akan digunakan.
2. Pilih data awal “p” secara acak.
3. Hitung jarak antara data “p” terhadap semua data menggunakan Euclidian distance.
4. Ambil semua amatan yang density-reachable dengan amatan “p”.
5. Jika amatan yang memenuhi nilai epsilon lebih dari jumlah minimal amatan dalam satu gerombol maka amatan “p” dikategorikan sebagai core points dan gerombol terbentuk.
6. Jika amatan “p” adalah border points dan tidak ada amatan yang density-reachable dengan amatan “p”, maka lanjutkan pada amatan lainnya.
7. Ulangi langkah 3 sampai 6 hingga semua amatan diproses.

UNBIN

Density Based Clustering

1. Klasterisasi berdasarkan kepadatan kumplan objek data
2. Kepadatan tidak selalu berbentuk bulat seperti lingkatan atau bola
3. Bisa berbentuk lain bahkan abstrak
4. Bagaimana mendefinisikan kepadatan?

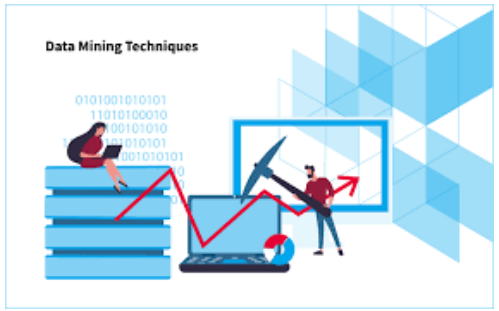




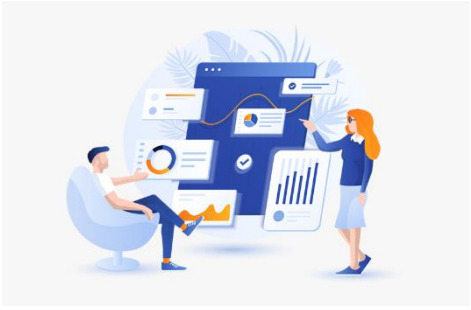
Characteristic DBSCAN

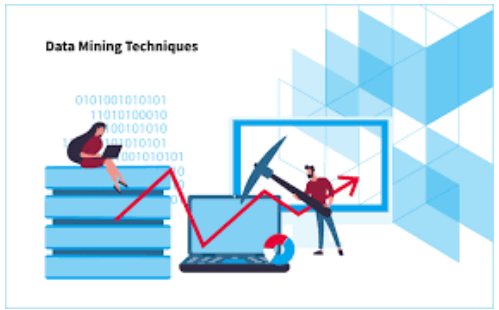
- Menemukan klaster berdasarkan area dengan kepadatan objek yang tinggi dan saling terhubung
- Kepadatan suatu objek diukur berdasarkan jumlah objek lain (minPts) yang berada di sekitarnya dalam radius tertentu.

UNBIN



CONTOH KASUS



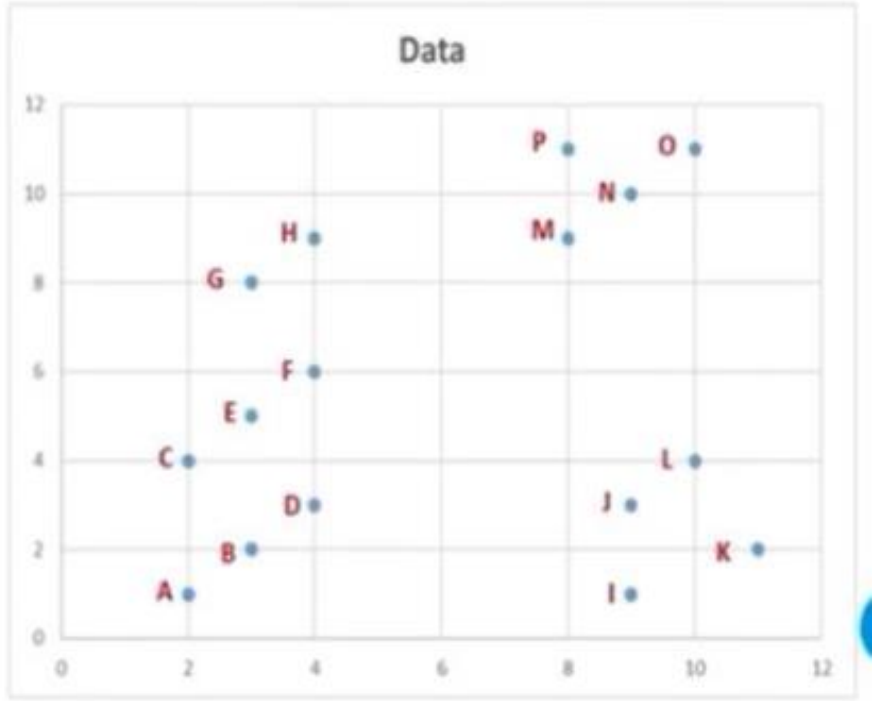


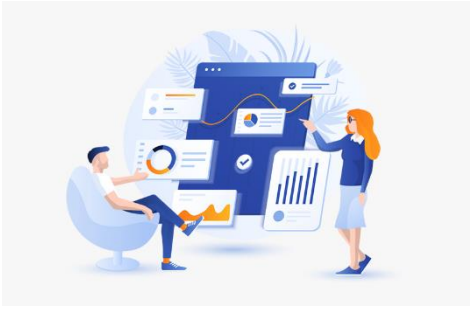
CONTOH KASUS : Lakukan klasterisasi menggunakan algoritma DBSCAN pada data ini

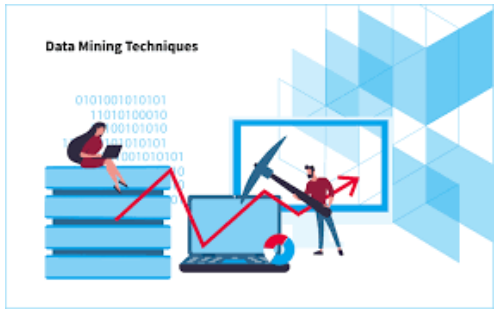
ID	X1	X2
A	2	1
B	3	2
C	2	4
D	4	3
E	3	5
F	4	6
G	3	8
H	4	9
I	9	1
J	9	3
K	11	2
L	10	4
M	8	9
N	9	10
O	10	11
P	8	11

Ditentukan nilai Epsilon dan MinPts sebagai berikut:

Epsilon	**3**
MinPts	4







Penyelesaian : Iterasi ke-1

ID	X1	X2
A	2	1
B	3	2
C	2	4
D	4	3
E	3	5
F	4	6
G	3	8
H	4	9
I	9	1
J	9	3
K	11	2
L	10	4
M	8	9
N	9	10
O	10	11
P	8	11

Jarak ke B	
A-B	1.414
B-B	0.000
C-B	2.236
D-B	1.414
E-B	3.000
F-B	4.123
G-B	6.000
H-B	7.071
I-B	6.083
J-B	6.083
K-B	8.000
L-B	7.280
M-B	8.602
N-B	10.000
O-B	11.402
P-B	10.296

Dalam Radius	
A	Ya
B	Ya
C	Ya
D	Ya
E	Ya
F	Ya
G	Tidak
H	Tidak
I	Tidak
J	Tidak
K	Tidak
L	Tidak
M	Tidak
N	Tidak
O	Tidak
P	Tidak

Tentukan titik pusat awal secara acak, misal : titik B

	Xp	Yp
B	3	2

Perhitungan jarak menggunakan ***Euclidean Distance*** ($\sqrt{(Xi-Xj)^2+(Yi-Yj)^2}$)

A ke B = $\sqrt{(2-3)^2+(1-2)^2}= 1.414$

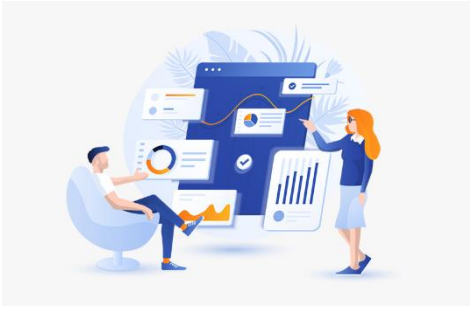
C ke B = $\sqrt{(2-3)^2+(4-2)^2}= 2.236$

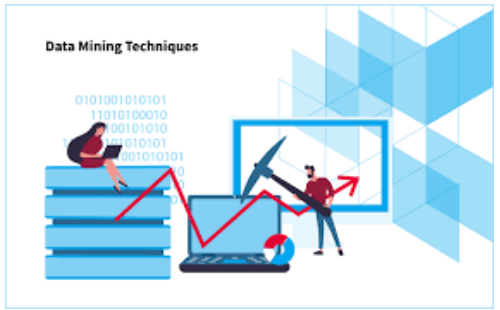
Dst.....

Total Ya :	5

Max:	3.000
Tidak Terpilih	E

Titik termasuk radius apabila nilai jarak <= Epsilon
Density Reachable = A, B, C, D, E
Neighboardhood Core Object terpenuhi karena jumlah e-neighborhood sudah memenuhi syarat yaitu Total Ya sudah >= minPts (4)
Titik selanjutnya dipilih berdasarkan jarak terjauh dari core object (E)



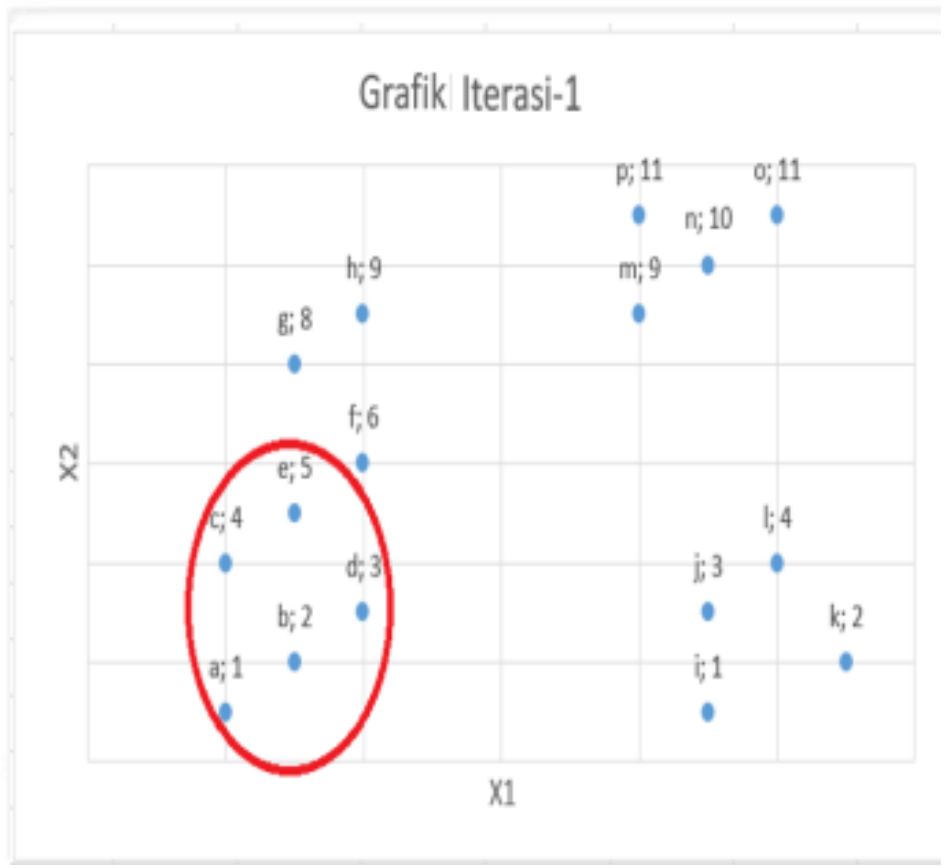


Penyelesaian : Iterasi ke-1

ID	X1	X2
A	2	1
B	3	2
C	2	4
D	4	3
E	3	5
F	4	6
G	3	8
H	4	9
I	9	1
J	9	3
K	11	2
L	10	4
M	8	9
N	9	10
O	10	11
P	8	11

Jarak ke B	
A-B	1.414
B-B	0.000
C-B	2.236
D-B	1.414
E-B	3.000
F-B	4.123
G-B	6.000
H-B	7.071
I-B	6.083
J-B	6.083
K-B	8.000
L-B	7.280
M-B	8.602
N-B	10.000
O-B	11.402
P-B	10.296

Dalam Radius	
A	Ya
B	Ya
C	Ya
D	Ya
E	Ya
F	Ya
G	Tidak
H	Tidak
I	Tidak
J	Tidak
K	Tidak
L	Tidak
M	Tidak
N	Tidak
O	Tidak
P	Tidak



UNBIN

Penyelesaian : Iterasi ke-2

→ Titik pusat kedua yang didapat dari iterasi pertama adalah titik E

	Xp	Yp
E	3	5

ID	X1	X2	Jarak ke E		Dalam Radius	
A	2	1	A-E	4.123	A	Tidak
B	3	2	B-E	3.000	B	Ya
C	2	4	C-E	1.414	C	Ya
D	4	3	D-E	2.236	D	Ya
E	3	5	E-E	0.000	E	Ya
F	4	6	F-E	1.414	F	Ya
G	3	8	G-E	3.000	G	Ya
H	4	9	H-E	4.123	H	Tidak
I	9	1	I-E	7.211	I	Tidak
J	9	3	J-E	6.325	J	Tidak
K	11	2	K-E	8.544	K	Tidak
L	10	4	L-E	7.071	L	Tidak
M	8	9	M-E	6.403	M	Tidak
N	9	10	N-E	7.810	N	Tidak
O	10	11	O-E	9.220	O	Tidak
P	8	11	P-E	7.810	P	Tidak

→ Hitung jarak menggunakan ***Euclidean Distance***

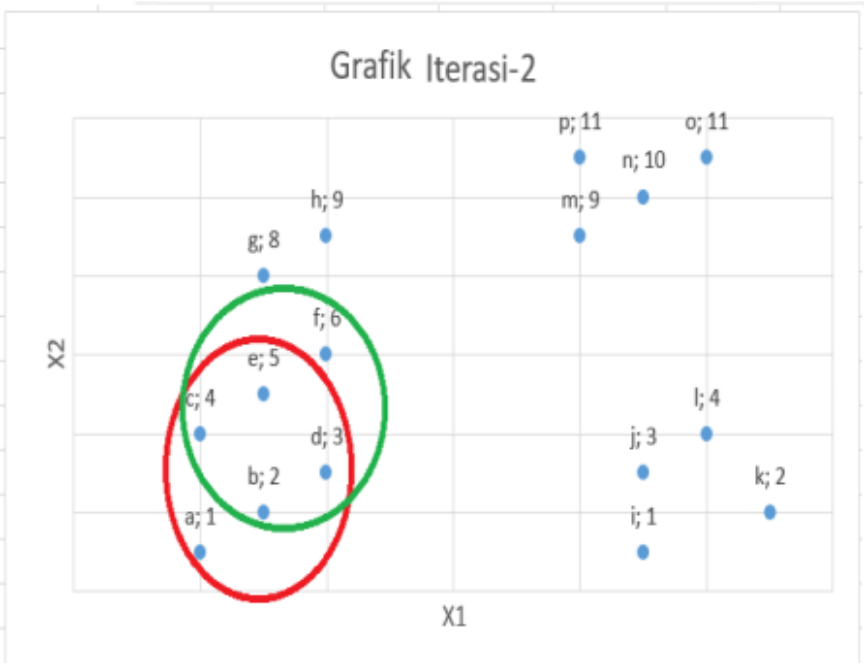
Diperoleh hasil:

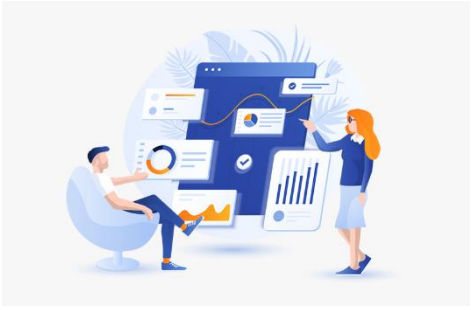
Total Ya :	6

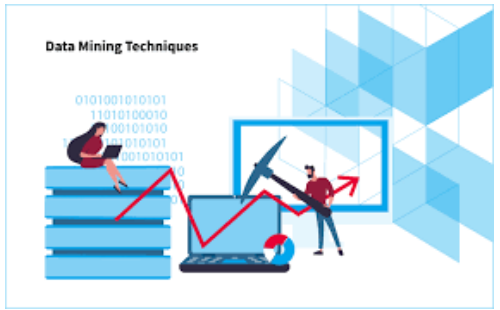
Max:	3.000
Tidak Terpilih	G

Density Reachable:
B, C, D, E, F, G

Titik selanjutnya yang dipilih yaitu titik G







Penyelesaian : Iterasi ke-3

→ Titik pusat ketiga yang didapat dari iterasi kedua adalah titik G

	Xp	Yp
G	3	8

ID	X1	X2	Jarak ke G		Dalam Radius	
A	2	1	A-G	7.071	A	Tidak
B	3	2	B-G	6.000	B	Tidak
C	2	4	C-G	4.123	C	Tidak
D	4	3	D-G	5.099	D	Tidak
E	3	5	E-G	3.000	E	Ya
F	4	6	F-G	2.236	F	Ya
G	3	8	G-G	0.000	G	Ya
H	4	9	H-G	1.414	H	Ya
I	9	1	I-G	9.220	I	Tidak
J	9	3	J-G	7.810	J	Tidak
K	11	2	K-G	10.000	K	Tidak
L	10	4	L-G	8,062	L	Tidak
M	8	9	M-G	5.099	M	Tidak
N	9	10	N-G	6.325	N	Tidak
O	10	11	O-G	7.616	O	Tidak
P	8	11	P-G	5.831	P	Tidak

→ Hitung jarak menggunakan ***Euclidean Distance***

Diperoleh hasil:

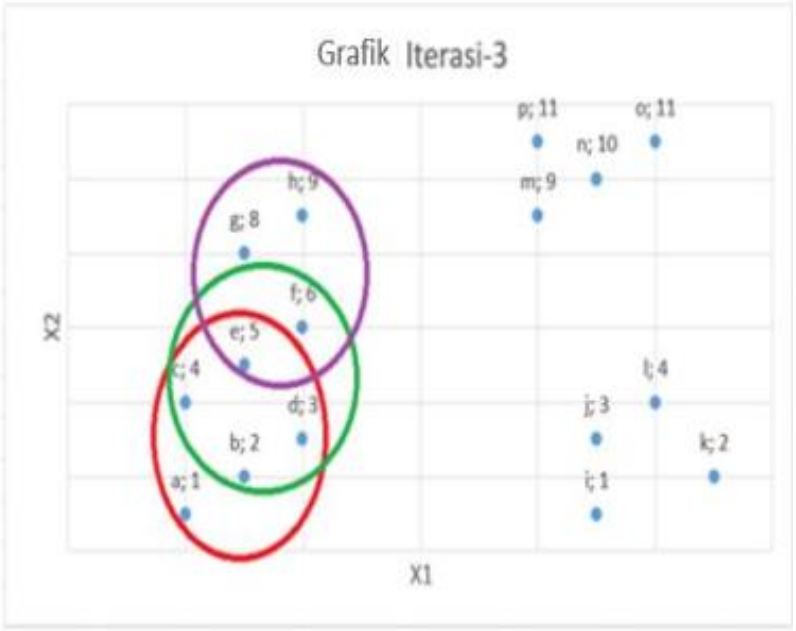
Total Ya :	4

Density Reachable:
E, F, G, H

Titik terjauh yang didapat yaitu titik E, akan tetapi tidak dapat dipilih karena sudah pernah digunakan.

Dan titik terjauh selanjutnya yaitu F tidak dapat dipilih karena termasuk dalam *border point* di *core object* pada proses sebelumnya.

Pada proses ini tidak ada titik yang dapat dijadikan *core object*



UNBIN

Penyelesaian : Iterasi ke-4

→ Titik pusat keempat ditentukan lagi secara acak yaitu titik J

	Xp	Yp
G	9	3

ID	X1	X2	Jarak ke G		Dalam Radius	
A	2	1	A-G	7.280	A	Tidak
B	3	2	B-G	6.083	B	Tidak
C	2	4	C-G	7.071	C	Tidak
D	4	3	D-G	5.000	D	Tidak
E	3	5	E-G	6.325	E	Tidak
F	4	6	F-G	5.831	F	Tidak
G	3	8	G-G	7.810	G	Tidak
H	4	9	H-G	7.810	H	Tidak
I	9	1	I-G	2.000	I	Ya
J	9	3	J-G	0.000	J	Ya
K	11	2	K-G	2.236	K	Ya
L	10	4	L-G	1.414	L	Ya
M	8	9	M-G	6.083	M	Tidak
N	9	10	N-G	7.000	N	Tidak
O	10	11	O-G	8.062	O	Tidak
P	8	11	P-G	8.062	P	Tidak

→ Hitung jarak menggunakan ***Euclidean Distance***

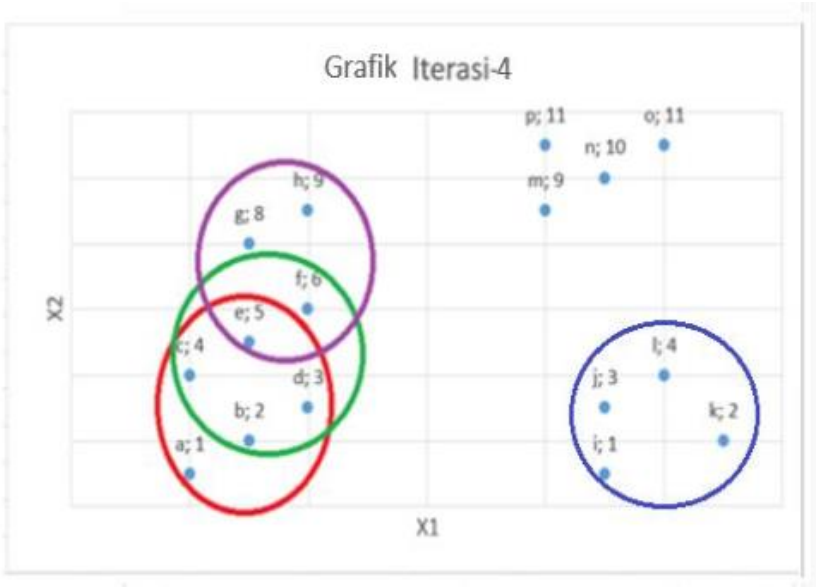
Diperoleh hasil:

Total Ya :	4

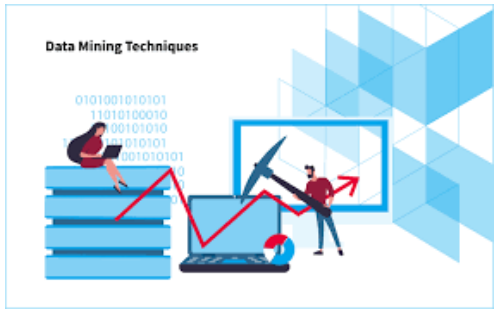
Max:	2.236
Tidak Terpilih	K

Density Reachable:
I, J, K, L

Titik terjauh yang didapat yaitu titik K



UNBIN



Penyelesaian : Iterasi ke-5

ID	X1	X2	Jarak ke K		Dalam Radius	
A	2	1	A-K	9.055	A	Tidak
B	3	2	B-K	8.000	B	Tidak
C	2	4	C-K	9.220	C	Tidak
D	4	3	D-K	7.071	D	Tidak
E	3	5	E-K	8.544	E	Tidak
F	4	6	F-K	8.062	F	Tidak
G	3	8	G-K	10.000	G	Tidak
H	4	9	H-K	9.899	H	Tidak
I	9	1	I-K	2.236	I	Ya
J	9	3	J-K	2.236	J	Ya
K	11	2	K-K	0.000	K	Ya
L	10	4	L-K	2.236	L	Ya
M	8	9	M-K	7.616	M	Tidak
N	9	10	N-K	8.246	N	Tidak
O	10	11	O-K	9.055	O	Tidak
P	8	11	P-K	9.487	P	Tidak

→ Titik pusat kelima yaitu titik K

	Xp	Yp
K	11	2

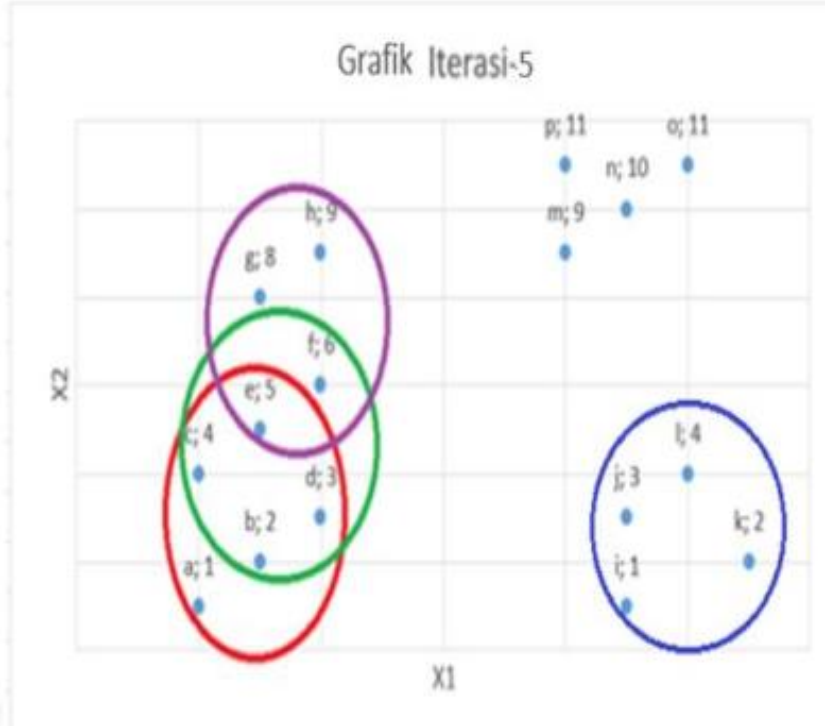
→ Hitung jarak menggunakan ***Euclidean Distance***

Diperoleh hasil:

Total Ya :	4

Density Reachable:
I, J, K, L

Karena titik density reachable yang didapat sama seperti iterasi sebelumnya, maka tidak ada titik yang dipilih kembali



UNBIN

Penyelesaian : Iterasi ke-6

ID	X1	X2	Jarak ke N		Dalam Radius	
A	2	1	A-N	11.402	A	Tidak
B	3	2	B-N	10.000	B	Tidak
C	2	4	C-N	9.220	C	Tidak
D	4	3	D-N	8.602	D	Tidak
E	3	5	E-N	7.810	E	Tidak
F	4	6	F-N	6.403	F	Tidak
G	3	8	G-N	6.325	G	Tidak
H	4	9	H-N	5.099	H	Tidak
I	9	1	I-N	9.000	I	Ya
J	9	3	J-N	7.000	J	Ya
K	11	2	K-N	8.246	K	Ya
L	10	4	L-N	6.083	L	Ya
M	8	9	M-N	1.414	M	Ya
N	9	10	N-N	0.000	N	Ya
O	10	11	O-N	1.414	O	Ya
P	8	11	P-N	1.414	P	Ya

→ Titik pusat keenam dipilih secara acak yaitu titik N

	Xp	Yp
N	9	10

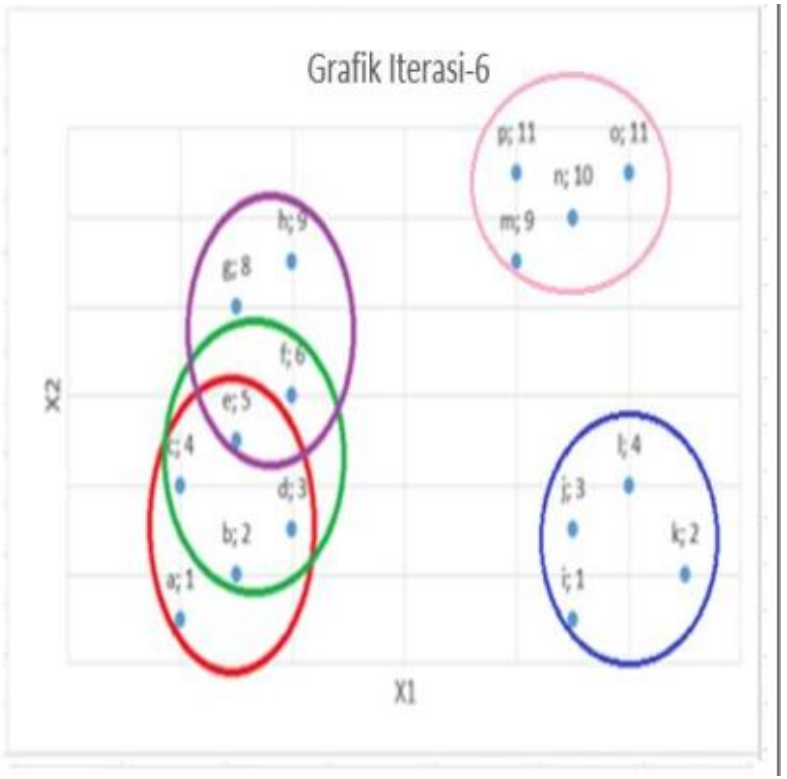
→ Hitung jarak menggunakan ***Euclidean Distance***

Diperoleh hasil:

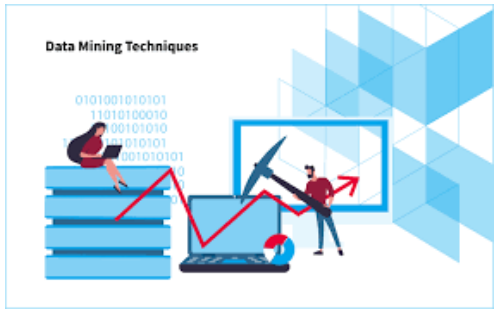
Total Ya :	4

Density Reachable:
M, N, O, P

Hasil sama seperti iterasi ke 5, maka tidak ada titik yang dipilih dan proses selesai pada iterasi ini.

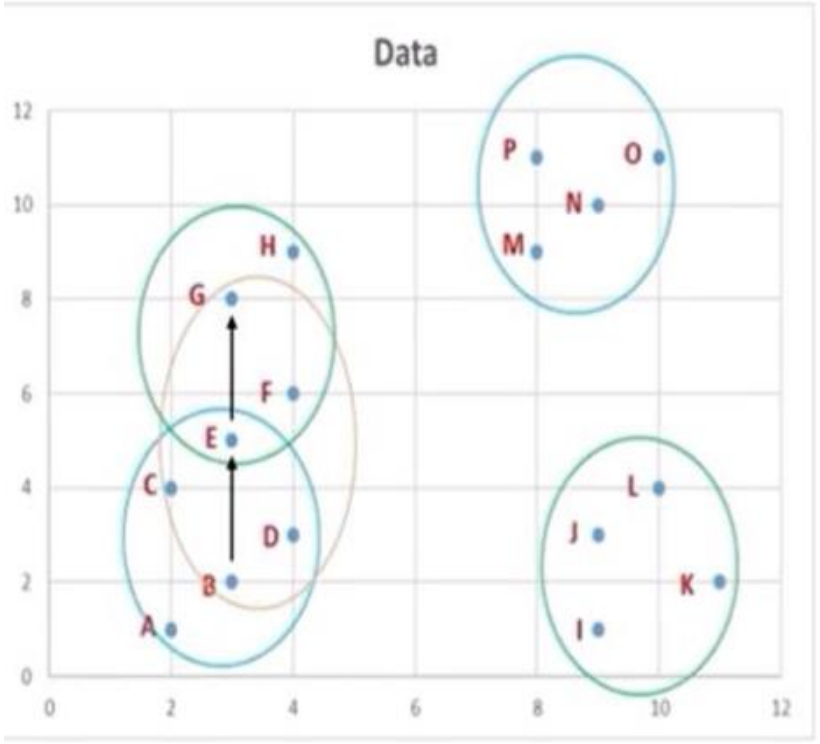


UNBIN



Penyelesaian : Clustering

Mendefinisikan hubungan antara *core object* yang merupakan sekumpulan *Neighborhood* menjadi satu *cluster*

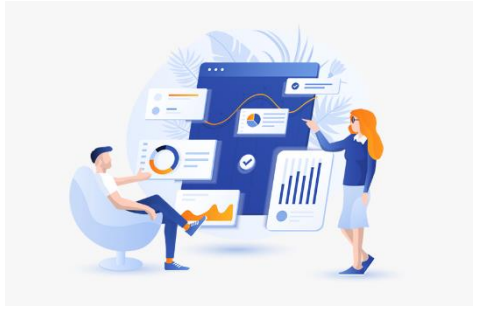


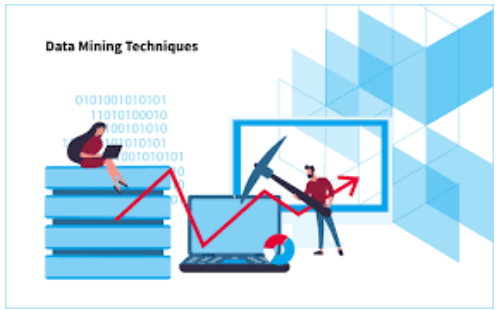
Coer point = B, E, G, J, N
Persyaratan hubungan = *core point* awal dengan akhir harus saling berhubungan *(density-connected)*

B density-connected terhadap G dimana telah melalui density-reachable terhadap titik E.

Sedangkan B dengan E merupakan hubungan *directly density-reachable*

Tidak ada data noise karena selutuh titik masuk ke dalam bagian cluster





HASIL AKHIR CLUSTER