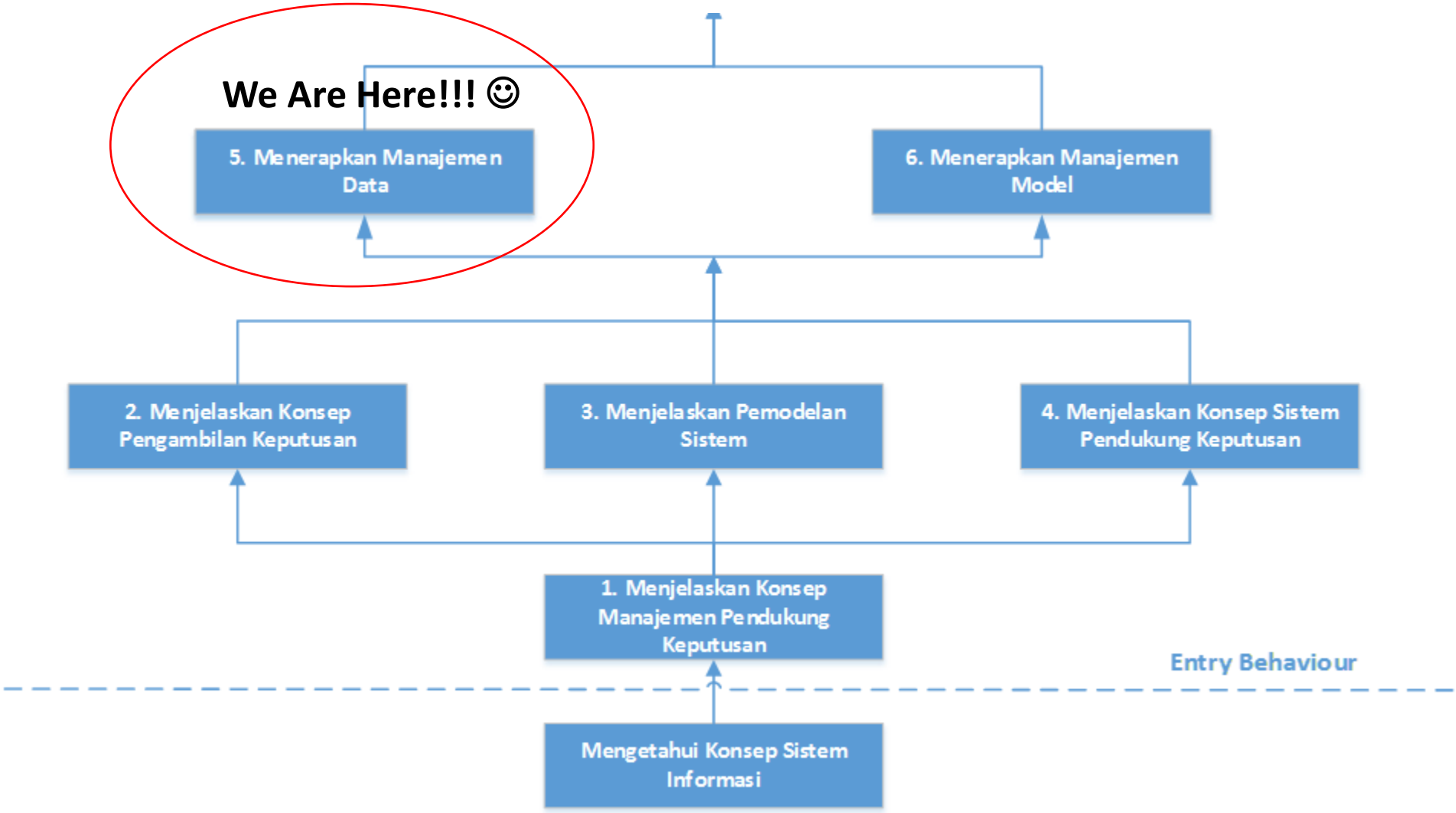
Manajemen Data

DECISION SUPPORT SYSTEM [D10K-5B01]

Sub Capaian Pembelajaran MK



AGENDA

1. Pendahuluan Subsistem Manajemen Data
2. Model Data Mining
3. Klasifikasi
4. Clustering

Clustering

What is Cluster Analysis?

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters

- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns

Major Clustering Approaches 1

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches 2

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus

Partitioning Methods

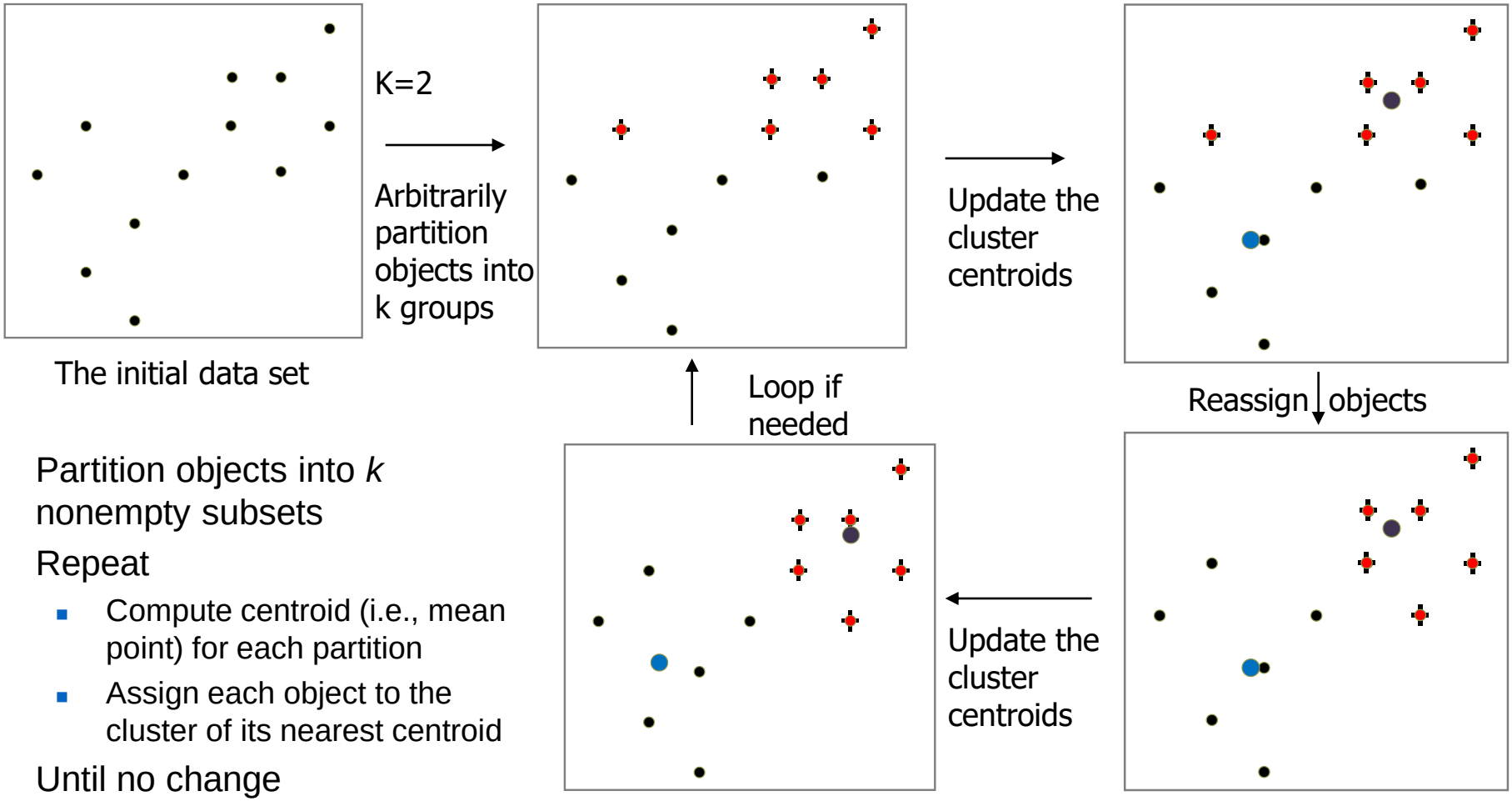
Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (d(p, c_i))^2$$

- Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

An Example of K-Means Clustering



- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Tahapan Algoritma k-Means

1. Pilih jumlah klaster k yang diinginkan
2. Inisialisasi k pusat klaster (centroid) secara random
3. Tempatkan setiap data atau objek ke klaster terdekat. Kedekatan dua objek ditentukan berdasar jarak. Jarak yang dipakai pada algoritma k-Means adalah *Euclidean distance* (*d*)

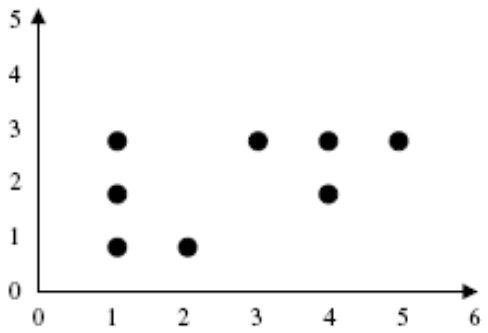
$$d_{Euclidean}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

   - x = x1, x2, . . . , xn, dan y = y1, y2, . . . , yn merupakan banyaknya *n* atribut(kolom) antara 2 record

4. Hitung kembali pusat klaster dengan keanggotaan klaster yang sekarang. Pusat klaster adalah rata-rata (mean) dari semua data atau objek dalam klaster tertentu
5. Tugaskan lagi setiap objek dengan memakai pusat klaster yang baru. Jika pusat klaster sudah tidak berubah lagi, maka proses pengklasteran selesai. Atau, kembali lagi ke langkah nomor 3 sampai pusat klaster tidak berubah lagi (stabil) atau tidak ada penurunan yang signifikan dari nilai SSE (*Sum of Squared Errors*)

Contoh Kasus – Iterasi 1

Instances	X	Y
A	1	3
B	3	3
C	4	3
D	5	3
E	1	2
F	4	2
G	1	1
H	2	1



1. Tentukan jumlah klaster k=2
2. Tentukan centroid awal secara acak misal dari data disamping m1 =(1,1), m2=(2,1)
3. Tempatkan tiap objek ke klaster terdekat berdasarkan nilai centroid yang paling dekat selisihnya (jaraknya). Didapatkan hasil, anggota *cluster1 = {A,E,G}, cluster2={B,C,D,F,H}*

   Nilai SSE yaitu:

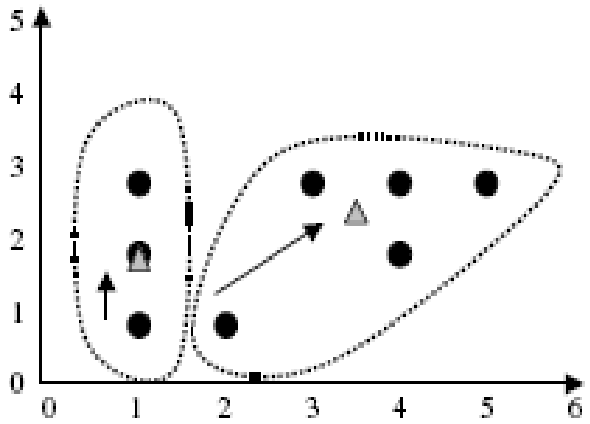
$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2$$

$$2^2 + 2{,}24^2 + 2{,}83^2 + 3{,}61^2 + 1^2 + 2{,}24^2 + 0^2 + 0^2 = 36$$

Interasi 2

Instances	X	Y
A	1	3
B	3	3
C	4	3
D	5	3
E	1	2
F	4	2
G	1	1
H	2	1

Point	Distance from m_1	Distance from m_2	Cluster Membership
a	2.00	2.24	C_1
b	2.83	2.24	C_2
c	3.61	2.83	C_2
d	4.47	3.61	C_2
e	1.00	1.41	C_1
f	3.16	2.24	C_2
g	0.00	1.00	C_1
h	1.00	0.00	C_2



4. Menghitung nilai centroid yang baru

$$m_1 = [(1+1+1)/3, (3+2+1)/3] = (1,2)$$

$$m_2 = [(3+4+5+4+2)/5, (3+3+3+2+1)/5] = (3{,}6;2{,}4)$$

5. Tugaskan lagi setiap objek dengan memakai pusat klaster yang baru.

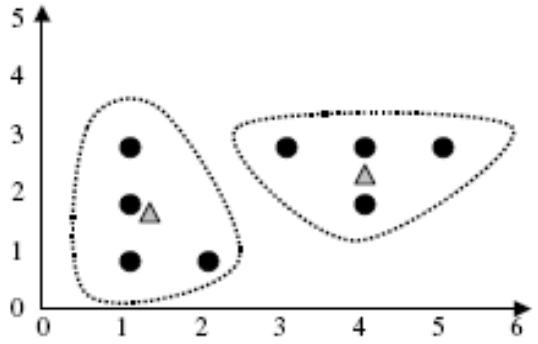
   Nilai SSE yang baru:

$$\text{SSE} = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2 = 1^2 + 0.85^2 + 0.72^2 + 1.52^2 + 0^2 + 0.57^2 + 1^2$$

$$+1.41^2 = 7.88$$

Iterasi 3

Point	Distance from m_1	Distance from m_2	Cluster Membership
a	1.00	2.67	C_1
b	2.24	0.85	C_2
c	3.16	0.72	C_2
d	4.12	1.52	C_2
e	0.00	2.63	C_1
f	3.00	0.57	C_2
g	1.00	2.95	C_1
h	1.41	2.13	C_2



4. Terdapat perubahan anggota cluster yaitu cluster1={A,E,G,H}, cluster2={B,C,D,F}, maka cari lagi nilai centroid yang baru yaitu: m1=(1,25;1,75) dan m2=(4;2,75)
5. Tugaskan lagi setiap objek dengan memakai pusat klaster yang baru
   Nilai SSE yang baru:

$$\text{SSE} = \sum_{i=1}^{k} \sum_{p \in C_i} d(p,m_i)^2 = 1.27^2 + 1.03^2 + 0.25^2 + 1.03^2 + 0.35^2 + 0.75^2$$

$$+0.79^2 + 1.06^2 = 6.25$$

Hasil Akhir

Point	Distance from m_1	Distance from m_2	Cluster Membership
a	1.27	3.01	C_1
b	2.15	1.03	C_2
c	3.02	0.25	C_2
d	3.95	1.03	C_2
e	0.35	3.09	C_1
f	2.76	0.75	C_2
g	0.79	3.47	C_1
h	1.06	2.66	C_2

- Dapat dilihat pada tabel. Tidak ada perubahan anggota lagi pada masing-masing cluster
- Hasil akhir yaitu: cluster1={A,E,G,H}, dan cluster2={B,C,D,F}
  Dengan nilai SSE = 6,25 dan jumlah iterasi 3

Density-Based Methods

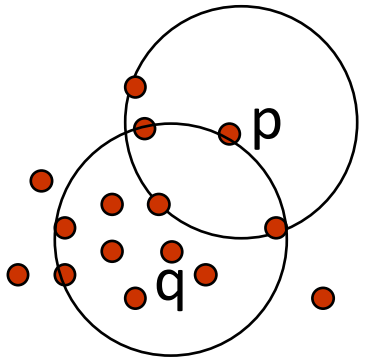
Density-Based Clustering Methods

- Pengelompokan berdasarkan kepadatan (kriteria cluster lokal), seperti titik yang terhubung dengan kepadatan
- Fitur utama:
  - Temukan kelompok dengan bentuk yang berubah-ubah
  - Dapat menangani noise
  - Satu pemindaian
  - Perlu parameter kepadatan sebagai kondisi terminasi
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

- Dua parameter*:*
  - *Eps: Radius maksimum lingkungan*
  - *MinPts: Jumlah minimum poin di lingkungan Eps dari titik itu*
- $N_{Eps}(q)$: {p belongs to D | dist(p,q) ≤ Eps}
- Directly density-reachable: Sebuah titik p secara langsung dapat dicapai dengan kerapatan dari titik q w.r.t. Eps, MinPts jika
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:

$$|N_{Eps}(q)| \geq MinPts$$

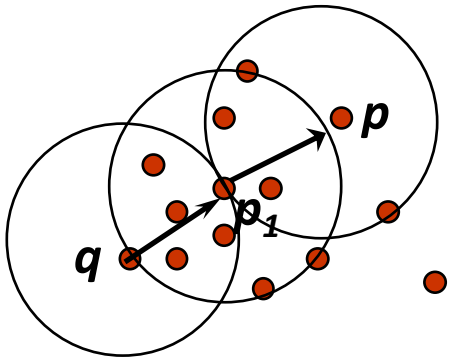


MinPts = 5

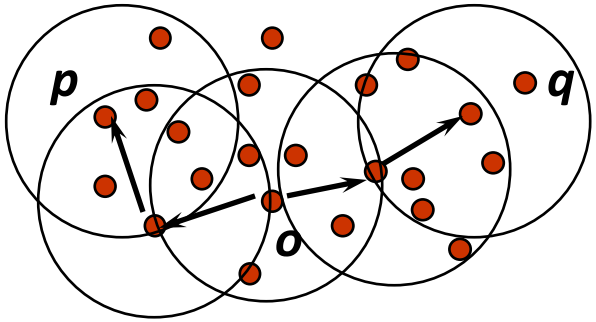
Eps = 1 cm

Density-Reachable and Density-Connected

- Density-reachable:
  - Titik p dapat dicapai dengan kerapatan dari titik q jika ada rantai titik p_1, …, p_n, $p_1 = q$, $p_n = p$ sedemikian rupa sehingga p_{i+1} dapat dicapai langsung oleh kerapatan dari p_i

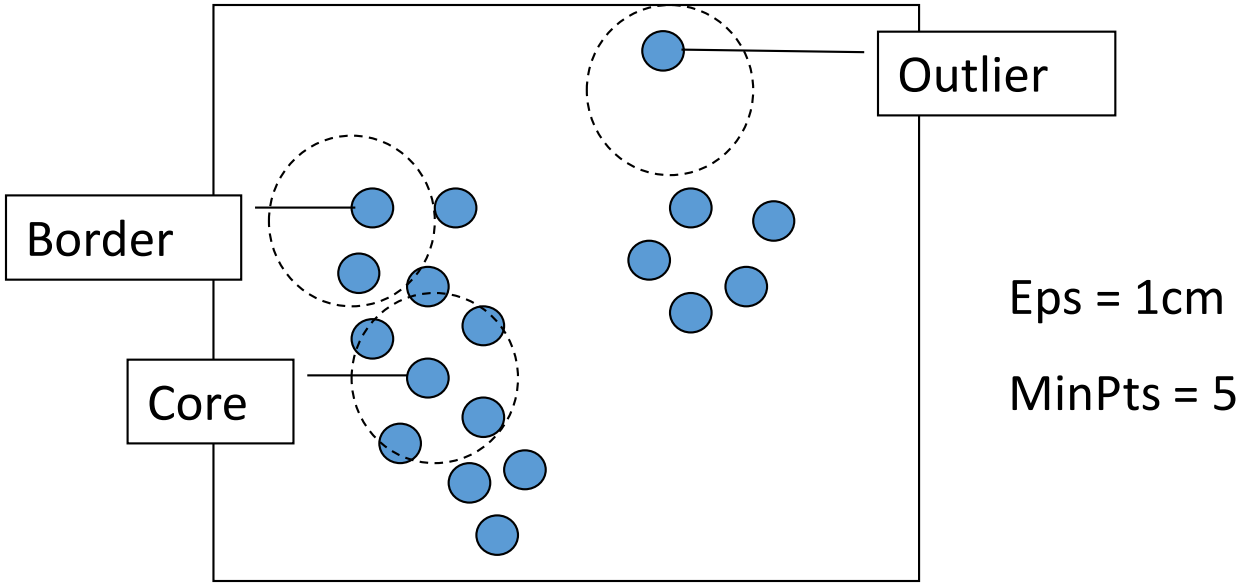


- Density-connected
  - Sebuah titik p terhubung dengan kepadatan ke titik q jika ada titik o sehingga keduanya, p dan q dapat dicapai kerapatan dari o



DBSCAN: Density-Based Spatial Clustering of Applications with Noise

- Menggunakan ide pembentukan cluster berbasis kepadatan: cluster didefinisikan sebagai kumpulan maksimal titik yang terhubung dengan kepadatan
- Menemukan cluster dengan berbagai bentuk dalam database spasial dengan adanya noise



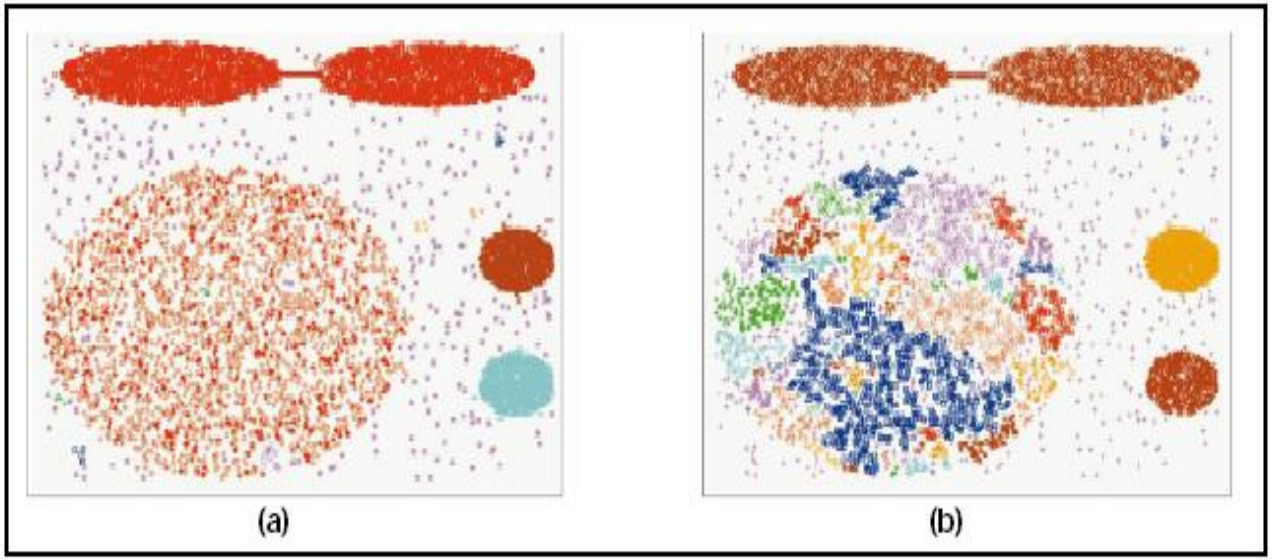
DBSCAN: The Algorithm

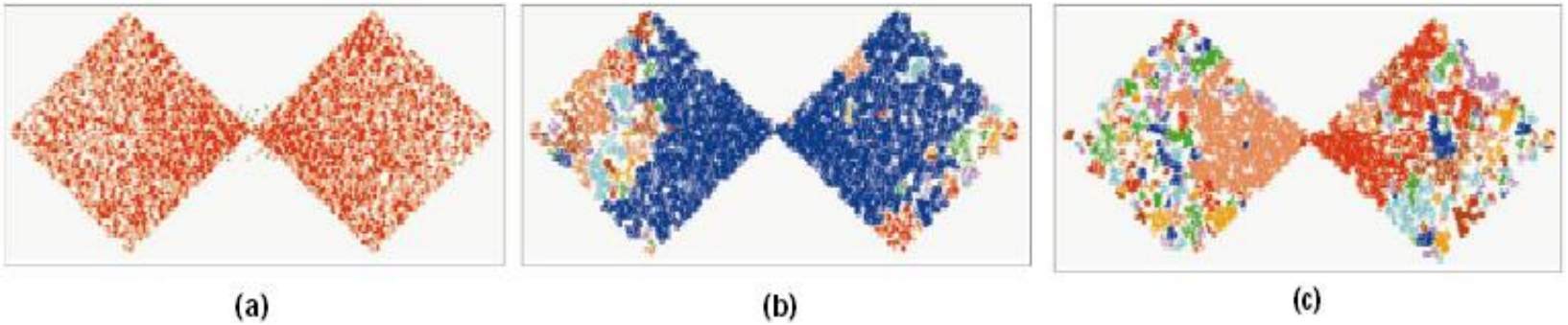
1. Pilih sembarang titik p
2. Ambil semua titik yang density-reachable dari p
3. Jika p adalah titik inti, maka terbentuk cluster
4. Jika p adalah titik batas, tidak ada titik yang density-reachable oleh i p dan DBSCAN mengunjungi titik database berikutnya
5. Lanjutkan proses hingga semua poin selesai diproses

Jika indeks spasial digunakan, kompleksitas komputasi DBSCAN adalah O (nlogn), di mana n adalah jumlah objek database. Jika tidak, kompleksitasnya adalah O (n2)

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

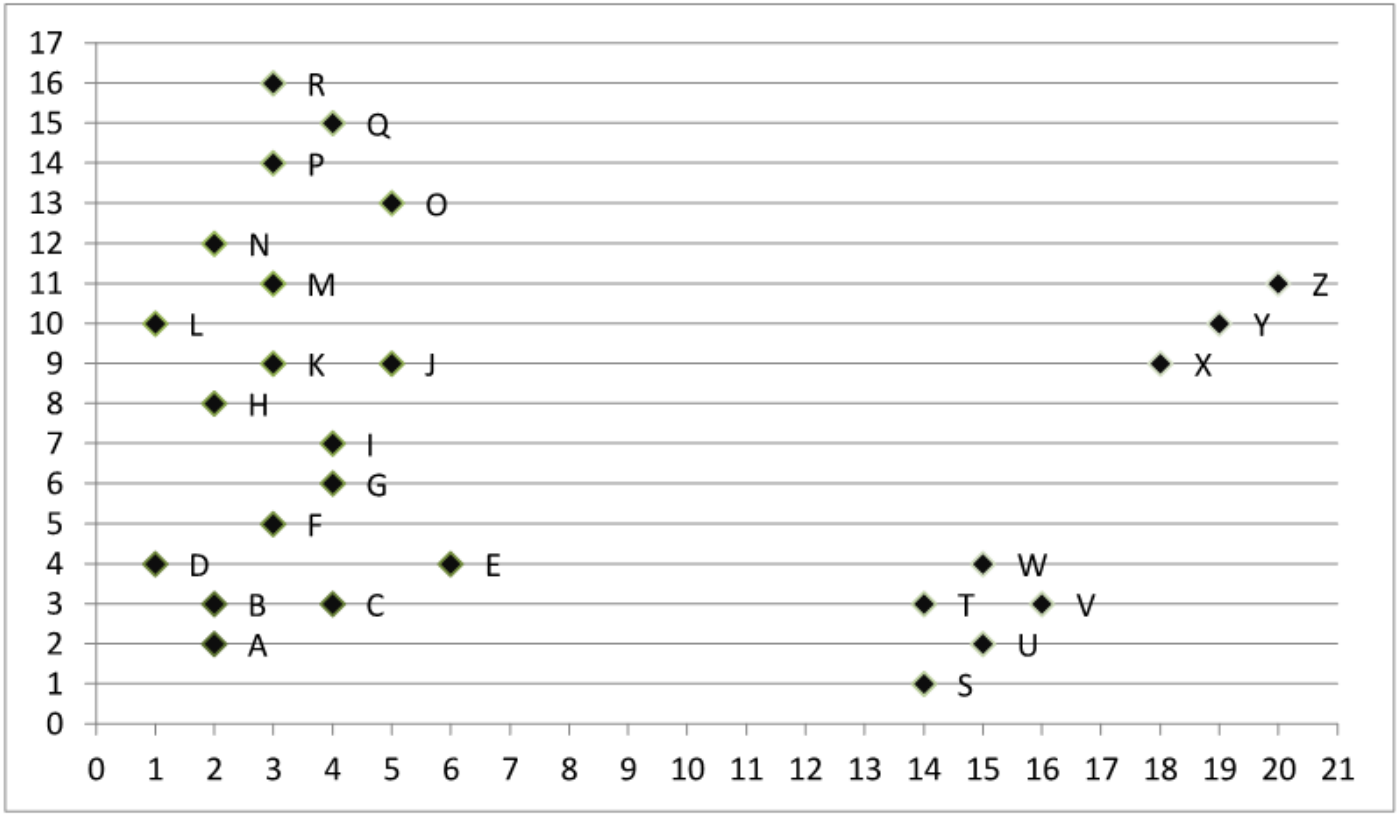




http://webdocs.cs.ualberta.ca/~yaling/Cluster/Applet/Code/Cluster.html

Soal DBSCAN

Titik	X	Y
A	2	2
B	2	3
C	4	3
D	1	4
E	6	4
F	3	5
G	4	6
H	2	8
I	4	7
J	5	9
K	3	9
L	1	10
M	3	11
N	2	12
O	5	13
P	3	14
Q	4	15
R	3	16
S	14	1
T	14	3
U	15	2
V	16	3
W	15	4
X	18	9
Y	19	10
Z	20	11



Dengan parameter input

a. MinPts : 5
b. Eps : 4

Iterasi 1

- Misal titik B sebagai pusat
- Hit jarak masing masing titik terhadap titik pusat B (mis. Dengan Euclidean Distance)

$$AB = \sqrt{(2-2)^2 + (2-3)^2} = 1$$

NB:
Dilakukan hal yang sama untuk titik yang lain

Jarak	Hasil
AB	1
BB	0
CB	2
DB	1,414214
EB	4,123106
FB	2,236068
GB	3,605551
HB	5
IB	4,472136
JB	6,708204
KB	6,082763
LB	7,071068
MB	8,062258
NB	9
OB	10,44031
PB	11,04536
QB	12,16553
RB	13,0384
SB	12,16553
TB	12
UB	13,0384
VB	14
WB	13,0384
XB	17,08801
YB	18,38478
ZB	19,69772

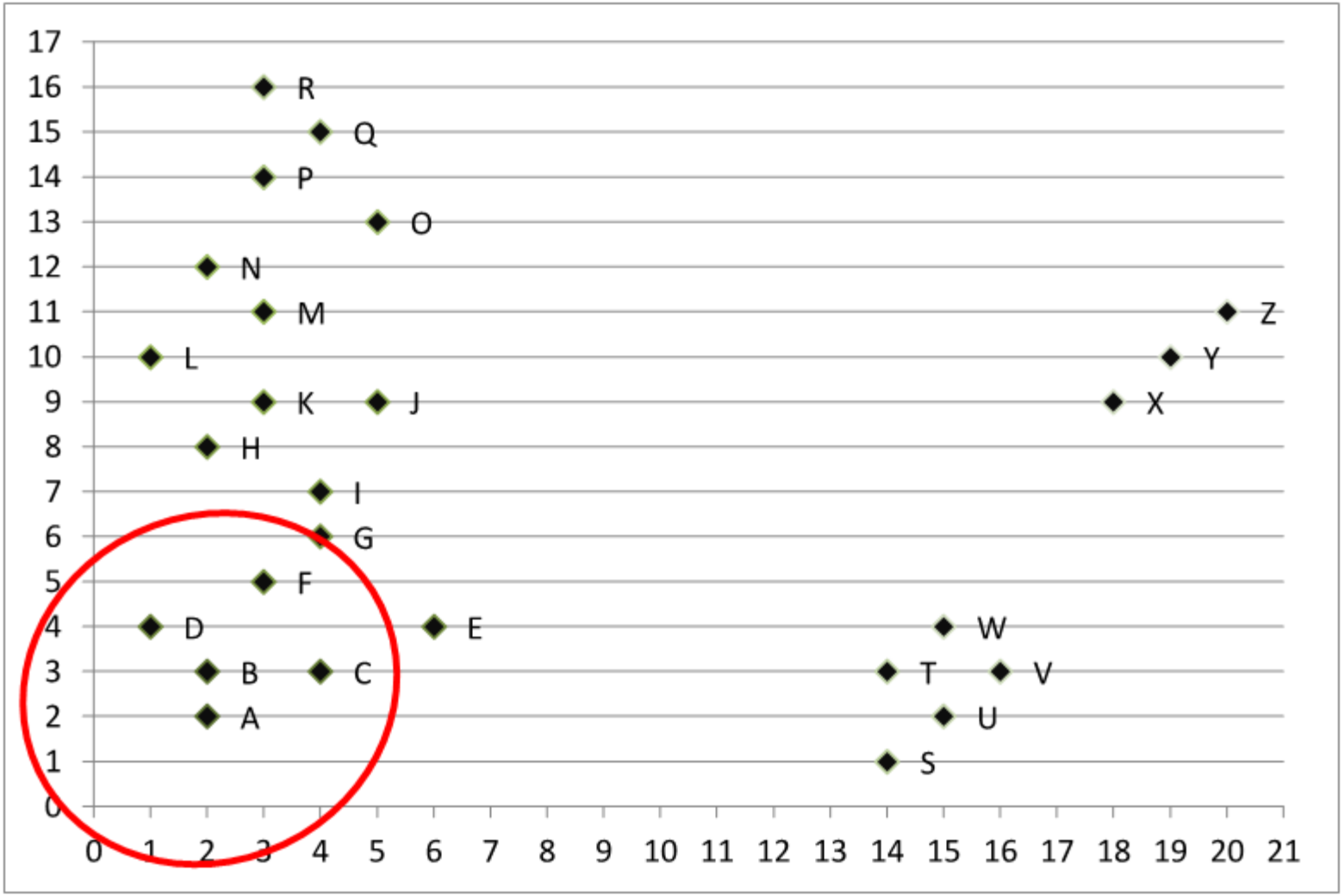
Pengambilan titik yang density reachable

- Ambil semua point yang density reachable terhadap titik pusatnya. Karena Eps=4 maka nilai titik yang memenuhi syarat adalah

titik	X	Y
A	2	2
B	2	3
C	4	3
D	1	4
F	3	5
G	4	6

Dari jumlah titik yang terpilih tersebut yaitu berjumlah 6. Jumlah ini sudah memenuhi untuk terbentuknya neighborhood core object karena jumlah objek e- neighborhood sudah memenuhi jumlah MinPts=5

Hasil iterasi 1



Iterasi 2

- Pilih titik yang memiliki jarak terjauh yang masih termasuk dalam dari core object pada iterasi pertama.

titik	X	Y	Jarak Ke titik B
A	2	2	1
B	2	3	0
C	4	3	2
D	1	4	1,414213562
F	3	5	2,236067977
G	**4**	**6**	**3,605551275**

Iterasi 2 (lanjutan)

1. hitung jarak masing-masing titik dengan core point untuk iterasi kedua,

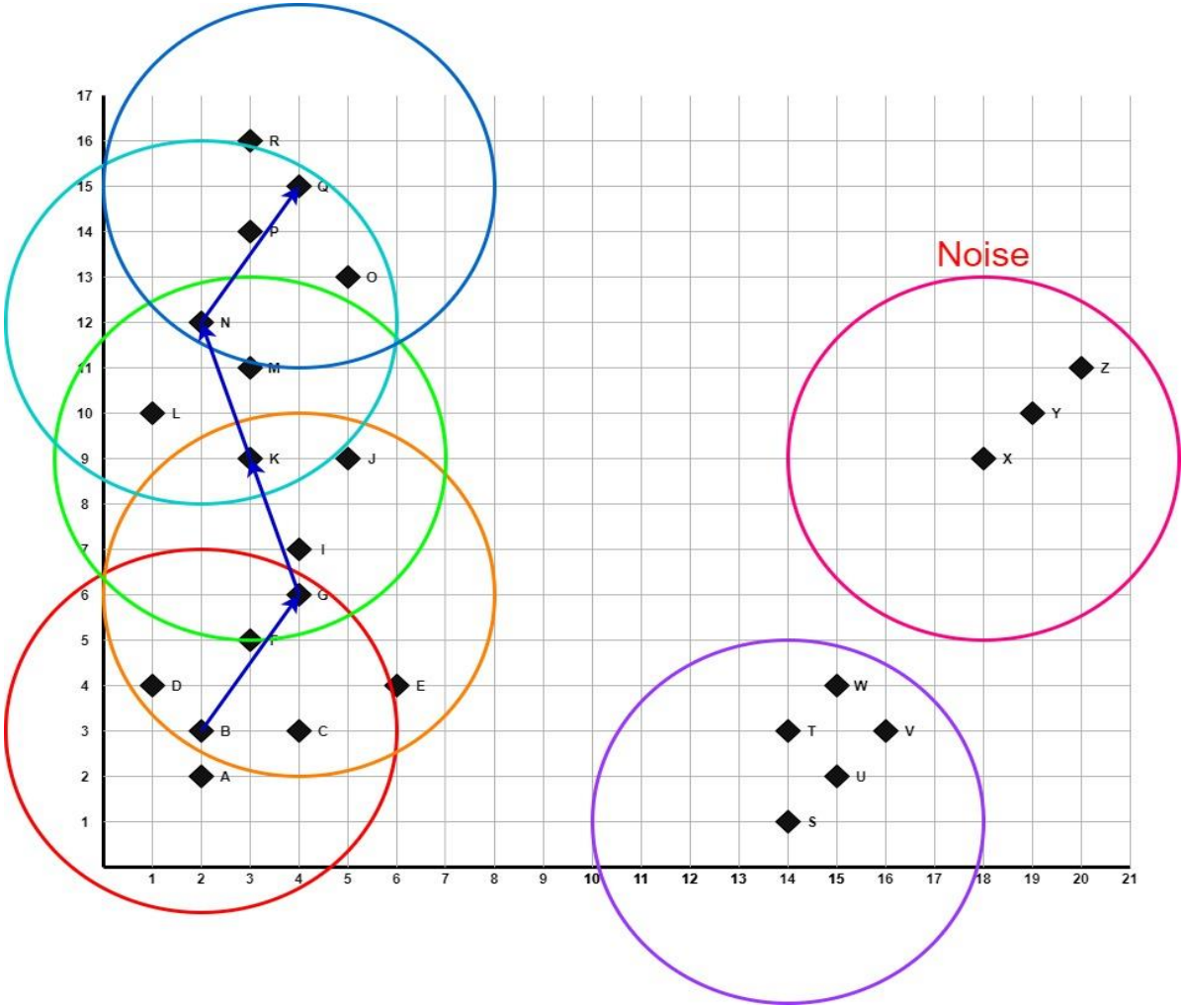
Jarak	Hasil
AG	4,472136
BG	3,605551
CG	3
DG	3,605551
EG	2,828427
FG	1,414214
GG	0
HG	2,828427
IG	1
JG	3,162278
KG	3,162278
LG	5
MG	5,09902
NG	6,324555
OG	7,071068
PG	8,062258
QG	9
RG	10,04988
SG	11,18034
TG	10,44031
UG	11,7047
VG	12,36932
WG	11,18034
XG	14,31782
YG	15,52417
ZG	16,76305

2. Ambil semua point yang density reachable terhadap titik pusatnya.

titik	X	Y
B	2	3
C	4	3
D	1	4
E	6	4
F	3	5
G	4	6
H	2	8
I	4	7
J	5	9
K	3	9

Dari jumlah titik yang terpilih adalah 10. Jumlah ini sudah memenuhi untuk terbentuknya neighborhood core object

Hasil Akhir



Note

Contoh soal dan penyelesaian K-means clustering dan DBSCAN disediakan di regular.live.unpad.ac.id