Kuliah 5
Deteksi Anomali

Referensi :

1. Tan P., Michael S., & Vipin K. 2006. *Introduction to Data mining*. Pearson Education, Inc. – Chapter 10
2. Han J & Kamber M. 2006. *Data mining – Concept and Techniques*. 2nd Edition, Morgan-Kauffman, San Diego - Chapter 7

Deteksi Anomali/Pencilan (Outlier)

- Apa itu Anomali/Pencilan (outlier)?
  - Sutu himpunan atau sekumpulan titik data yang sangat berbeda (different - Tan) / sangat tidak mirip (dissimilar-Han) dengan data lainnya
- Bebeberapa masalah pendeteksian anomali/pencilan (outlier)
  - Diberikan sebuah basis data D, temukan semua titik data $\mathbf{x} \in D$ dengan skor anomali lebih besar dari suatu nilai ambang (threshold) t
  - Diberikan sebuah basis data D, temukan semua titik data $\mathbf{x} \in D$ yang setelah diurutkan termasuk ke dalam n titik data yang memiliki skor anomali terbesar (top-n largest anomaly scores f($\mathbf{x}$))
  - Diberikan sebuah basis data D, yang hampir seluruhnya berisi titik data yang normal tapi tanpa label (but unlabeled), dan sebuah titik uji **x**, hitung skor anomali **x** terhadap basis data D
- Aplikasi:
  - Deteksi kecurangan/penipuan pada kartu kredit (Credit card fraud detection,) deteksi penipuan pada telekomunikasi, deteksi *network intrusion*, deteksi kesalahan (fault detection) pada sistem (kondisi tidak normal)

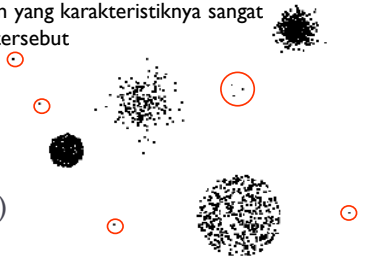
Deteksi anomali

- Tantangan
  - Berapa banyak pencilan (outliers) dalam data?
  - Metode yang digunakan adalah *unsupervised*
    - Validasi dapat menjadi tantangan tersendiri (seperti pada *clustering*)
  - "Menemukan jarum di tumpukan jerami"

- Asumsi penyelesaiannya:
  - Ada jauh lebih banyak pengamatan yang "normal" dari pada yang "abnormal" (pencilan/anomali) dalam data

Skema deteksi anomali

- Tahapan umum
  - Buat profil dari kondisi/karakteristik "normal"
    - Profil dapat berupa pola atau ringkasan secara statistik dari seluruh populasi
  - Gunakan profil "normal" ini untuk mendeteksi anomali
    - Anomali adaralh hasil pengamatan yang karakteristiknya sangat berbeda dengan profil "normal" tersebut

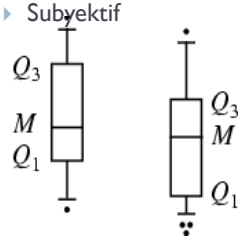


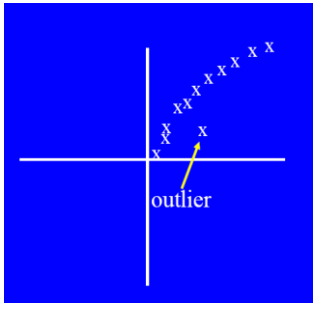
- Jenis-jenis skema deteksi anomali
  - Berbasis grafik dan statistik (Graphical & Statistical-based)
  - Berbasis jarak (Distance-based)
  - Berbasis model (Model-based)

Pendekatan berbasis grafik

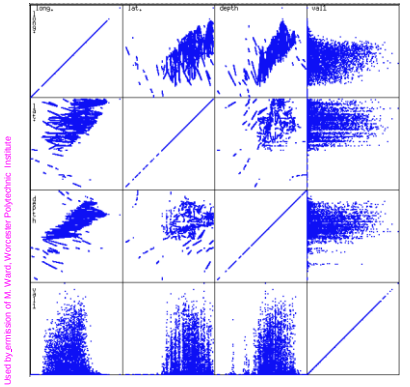
- Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)

- Keterbatasan
  - Memboroskan waktu (Time consuming)
  - Subyektif





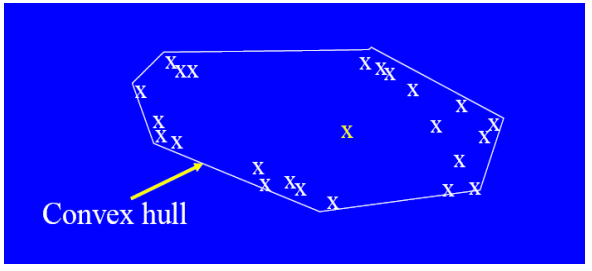
Matriks Scatterplot [Cleveland 93]



matrix of scatterplots (x-y-diagrams) of the k-dimensional data [total of (k2/2-k) scatterplots]

Metode Convex Hull

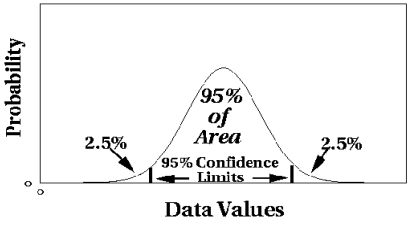
- Titik ekstrim diasumsikan sebagai pencilan/outliers
- Gunakan metode convex hull untuk mendeteksi nilai ekstrim



- Bagaimana jika ternyaa pencilan/outlier berada di tengah data?

Pendekatan statistika

- Asumsikan sebuah model parametrik yang mendeskripsikan distribusi data (Contoh: distribusi normal)
- Lakukan uji statistik sesuai dengan
  - Distribusi data
  - Parameter dari distribusi (misal: mean, variance)
  - Jumlah pencilan/outlier yang diharapkan (batas dari nilai-nilai selang kepercayaan/confidence limit)



Pendekatan berbasis jarak

- Metode ini diperkenalkan untuk mengatasi kelemahan/keterbatasan dari metode statistik
  - Diperlukan analisis multi-dimensi tanpa mengetahui distribusi dari datanya
- Pencilan berbasis jarak(Distance-based outlier):
  - Sebuah pencilan berbasis jarak/DB(p, D)-outlier adalah suatu titik data/objek O dalam dataset T sedemikian sehingga paling sedikit ada suatu bagian/fraksi p dari titik-titik data/objek-objek di T yang jaraknya dari O lebih besar dari D

Pendekatan berbasis jarak

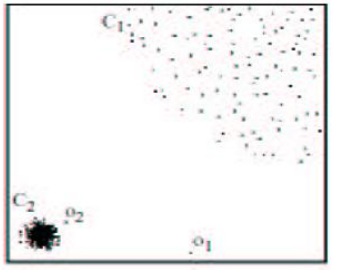
- Data direpresentasikan sebagai suatu vektor dari fitur (atribut)
- Tiga pendekatan utama
  - Berbasis ketetanggan terdekat (Nearest-neighbor)
  - Berbasis kepadatan (Density based)
  - Berbasis klustering (Clustering based)

Pendekatan berbasis ketetanggaan terdekat (Nearest-Neighbor)

- Pendekatan:
  - Hitung jarak tiap pasangan titik data
  - Ada berbagai cara untuk mendefinisikan pencilan/outlier:
    - Titik-titik data di mana titik tersebut hanya memiliki jumlah titik ketetanggaan (neighboring) yang lebih sedikit dari *p* dalam suatu jarak *D*
    - n titik data yang memiliki jarak tertinggi ke tetangga terdekat ke-k (k-th nearest neighbour)
    - n titik data yang memiliki jarak rata-rata tertinggi ke k tetangga terdekat (k nearest neighbors)

Deteksi pencilan lokal (local outlier) berbasis kepadatan (density)

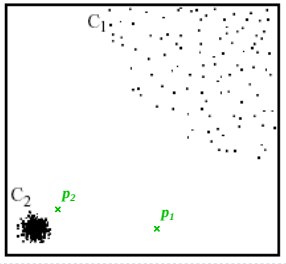
- Deteksi pencilan berbasis jarak adalah berdasarkan distribusi jarak global
- Sangat sulit mengidentifikasi pencilan/outlier jika data tidak terdistribusi secara searagam (uniform)
- Contoh C_1 terdiri atas 400 titik yang terdistribusi secara renggang (loosely distributed points), C_2 memiliki 100 titik-titik yang sangat rapat (tightly condensed points), 2 titik pencilan (outlier) o_1, o_2
- Metode berbasis jarak tidak dapat menidentifikasi o_2 sebagai pencilan
- → perlu mengenalkan konsep pencilan lokal (local outlier)



- Local outlier factor (LOF)
  - Asumsikan bahwa pencilan (outlier) tidak crisp
  - Tiap titik punya sebuah LOF

Berbasis kepadatan(Density-based): Pendekatan LOF

- Untuk tiap titik, hitung kepadatan dari ketetanggan lokalnya (local neighborhood)
- Hitung *local outlier factor* (LOF) dari sampel *p* sebagai nilai rata-rata dari rasion kepadatan (density) sampel *p* dan kepadatan (density) ketetanggan terdekatnya
- Pencilan (Outliers) adalah titik-titik yang memiliki nilai LOF terbesar



Pada pendekatan Nearest Neighbor, p_2 tidak diidentifikasi sebagai pencilan (outlier), namun dengan pendekatan LOF p_1 dan p_2 adalah pencilan (outlier)

Contoh LOF

- Diketahui 4 titik data :

  a(0,0), b(0,1), c(1,1), d(3,0)

- Hitung LOF untuk tiap titik dan perlihatkan top 1 outlier, tetapkan k = 2 dan gunakan jarak Manhattan

Tahap demi tahap LOF

Titik-titik: a(0,0), b(0,1), c(1,1), d(3,0)

- Tahap 1: Hitung jarak menggunakan jarak Manhattan

	a	b	c	d
a	-	1	2	3
b		-	1	4
c			-	3
d				-

- Tahap 2: Hitung $dist_2$ (o) (jarak tetangga terdekat ke-2 dari titik o)
  - $dist_2$ (a) = dist(a,c) = 2 (c adalah tetangga terdekat ke-2 dari a)
  - $dist_2$ (b) = dist(b,a) = 1 (a/c adalah tetangga terdekat ke-2 dari b)
  - $dist_2$ (c) = dist(c,a) = 2 (a adalah tetangga terdekat ke-2 dari c)
  - $dist_2$ (d) = dist(d,a) = 3 (a/c adalah tetangga terdekat ke-2 dari d)

Tahap demi tahap LOF

- Tahap 3: Hitung N_k(o)

$$N_k(o) = \{o' | \ o' \text{ dalam } D, \text{dist}(o, o') \leq \text{dist}_k(o)\}$$

N_k(o) adalah titik selain o dalam D di mana jarak antara o dan titik lainnya dalam D lebih kecil atau sama dengan jarak o dengan tetangga terdekat ke-2nya.

- $N_2(a) = \{b,c\} \rightarrow$ dist(a,b) = 1, dist(a,c)=2, $dist_2(a)$=2
- $N_2(b) = \{a,c\} \rightarrow$ dist(b,a) = 1, dist(b,c)=1, $dist_2(b)$=1
- $N_2(c) = \{b,a\} \rightarrow$ dist(c,b) = 1, dist(c,a)=2, $dist_2(c)$=2
- $N_2(d) = \{a,c\} \rightarrow$ dist(d,a) = 3, dist(d,c)=3, $dist_2(b)$=3

Tahap demi tahap LOF

- Tahap 4: Hitung lrd_k (o): Local Reachability Density of o

$$lrd_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)} \quad reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

$$lrd_k(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)}$$

Step by Step LOF

- $lrd_k(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)}$
- $reachdist_2(b \leftarrow a) = max\{dist_2(b), dist(b, a)\}$
  $= \max\{1, 1\} = 1$
- $reachdist_2(c \leftarrow a) = max\{dist_2(c), dist(c, a)\}$
  $= \max\{2, 2\} = 2$
- Thus, $lrd_2(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)} = 2/(1+2) = 0.667$
- Similarly.. $lrd_2(b) = \frac{\| N_2(b) \|}{reachdist_2(a \leftarrow b) + reachdist_2(c \leftarrow b)} = 2/(2+2) = 0.5$

  $lrd_2(c) = \frac{\| N_2(c) \|}{reachdist_2(b \leftarrow c) + reachdist_2(a \leftarrow c)} = 2/(1+2) = 0.667$

  $lrd_2(d) = \frac{\| N_2(b) \|}{reachdist_2(a \leftarrow d) + reachdist_2(c \leftarrow d)} = 2/(3+3) = 0.33$

Step by Step LOF

- Tahap 5: Hitung $LOF_k(o)$

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd_k(o')}{lrd_k(o)}}{\|N_k(o)\|} = \sum_{o' \in N_k(o)} lrd_k(o') \cdot \sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)$$

$LOF_2(a) = (lrd_2(b) + lrd_2(c)) * (reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a))$
$= (0.5+0.667) * (1+2) = 3.501$

$LOF_2(b) = (lrd_2(a) + lrd_2(c)) * (reachdist_2(a \leftarrow b) + reachdist_2(c \leftarrow b))$
$= (0.667+0.667) * (2+2) = 5.336$

$LOF_2(c) = (lrd_2(b) + lrd_2(a)) * (reachdist_2(b \leftarrow c) + reachdist_2(a \leftarrow c))$
$= (0.5+0.667) * (1+2) = 3.501$

$LOF_2(d) = (lrd_2(a) + lrd_2(c)) * (reachdist_2(a \leftarrow d) + reachdist_2(c \leftarrow d))$
$= (0.667+0.667) * (3+3) = 8.004$

Step by Step LOF

- Tahap 6: Urutkan semua $LOF_k(o)$
  - $LOF_2(d) = 8.004$
  - $LOF_2(b) = 5.336$
  - $LOF_2(a) = 3.501$
  - $LOF_2(c) = 3.501$

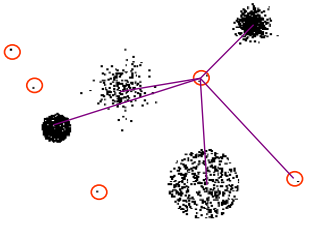
- Dapat dilihat dengan jelas, top 1 outlier adalah titik d (karena memiliki nilai LOF_2 tertinggi)

Pendekatan berbasis deviasi (Deviation-Based Approach)

- Identifikasi pencilan (outlier) dengan memeriksa karakteristik utama dari tiap objek dalam kelompok (grup)
- Objek-2 yang "menyimpang" dari deskripsi karakteristik tersebut dianggap sebagai pencilan/outlier
- Sequential exception technique
  - Simulasikan cara di mana manusia dapat membedakan objek yang tidak umum dari serangkaian objek-objek yang memiliki karakteristik seperti yang diharapkan
- OLAP data cube technique
  - Menggunakan data cubes untuk mengidentifikasi daerah anomali dalam data dengan multidimensi yang besar

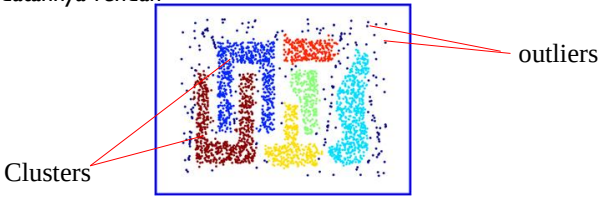
Berbasis klustering

- Ide dasar:
  - Kluster data ke dalam grup dengan kepadatan (density) yang berbeda
  - Pilih titik-titik dalam kluster kecil sebagai kandidat pencilan (outlier)
  - Hitung jarak antara titik yang menjadi kandidat pencilan dengan titik-titik kluster-kluster yang tidak mengandung kandidat pencilan (outlier)
    - Jika titik kandidat jauh dari semua titik yang bukan kandidat, maka titik tersebut adalah pencilan (outlier)



DBSCAN

- Density-based spatial clustering of application with noise (DBSCAN) adalah suatu algoritme klustering data yang diusulkan oleh Martin Ester, Hans-Peter Kriegel, Jörg Sander dan Xiaowei Xu pada tahun 1996.
- Metode ini mengelompokkan secara bersama titik-titik yang mengelompok secara dekat (closely packed together) (titik-titik yang memiliki banyak tetangga dekat/nearby neighbors)
- Tandai titik pencilan/outliers yang terletak sendirian di daerah yang kepadatannya rendah

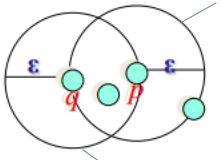


Definisi kepadatan (density)

- ε-Neighborhood – objek dengan radius ε dari suatu objek

$$N_{\varepsilon}(p) : \{q \mid d(p,q) \le \varepsilon\}$$

- "High density" - ε-Neighborhood dari objek yang terdiri atas paling sedikit **MinPts objek**



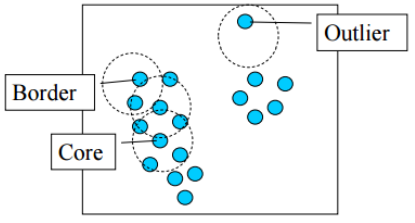
ε-Neighborhood dari *p*

MinPts=4

- Kepadatan/Density *p* tinggi
- Kepatan/Density *q* is "*rendah*"

ε-Neighborhood of *q*

Core, Border, & Outlier



ε = 1 unit, MinPts =5

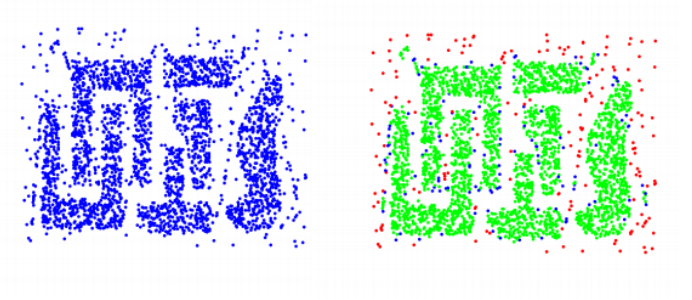
Diberikan ε dan MinPts, dikatagorisasikan objek-2 ke dalam tiga kelompok ekslusif: core point, border point, dan noise point

Suatu titik disebut sebagai core point jika ada lebih dari sejumlah titik (MinPts) dalam ε (Eps)—Titik-titik ini berada pada bagian dalam (interior) dari suatu kluster

Suatu border point memiliki lebih sedikit dari MinPts dalam ε (Eps), tetapi berada di dalam lingkungan dari *core point.*

Suatu noise point adalah titik bukan merpakan *core point* maupun *border point.*

Core, Border, & Outlier - Example



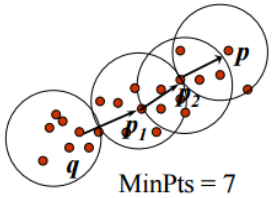
Original Points

Point types: core, border and outliers

ε = 10, MinPts = 4

Density-reachability

- Density-Reachable (secara langsung dan tidak langsung/directly and indirectly):
  - titik p directly density-reachable dari p_2
  - p_2 directly density-reachable dari p_1
  - p_1 directly density-reachable dari q
  - $p \leftarrow p_2 \leftarrow p_1 \leftarrow q$ membentuk suatu rantai



- p (indirectly) density-reachable dari q
- q is not density-reachable from p

Algotime DBSCAN

- Pilih sembarang titik *p*
- Temu kembalikan semua titik yang density-reachable dari *p* sesuai dengan ε (*Eps)* dan *MinPts*.
- Jika p adalah suatu core point, bentuk sebuah kluster
- Jika *p* adalah suatu border point, maka tidak ada titik-titik yang density-reachable dari *p* dan DBSCAN mengunjungi titik selanjutnya dalam basis data
- Ulangi proses sampai semua titik telah diproses

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Contoh DBSCAN

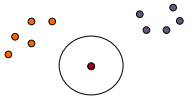
- Parameter
  - ε = 2 cm
  - *MinPts* = 3

for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Contoh DBSCAN

- Parameter
  - ε = 2 cm
  - *MinPts* = 3

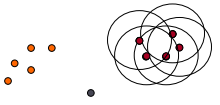


for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Contoh DBSCAN

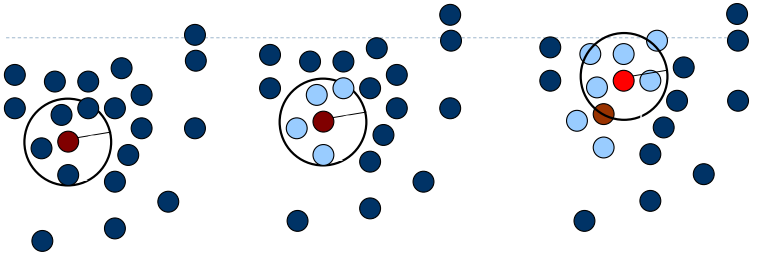
- Parameter
  - ε = 2 cm
  - *MinPts* = 3



for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

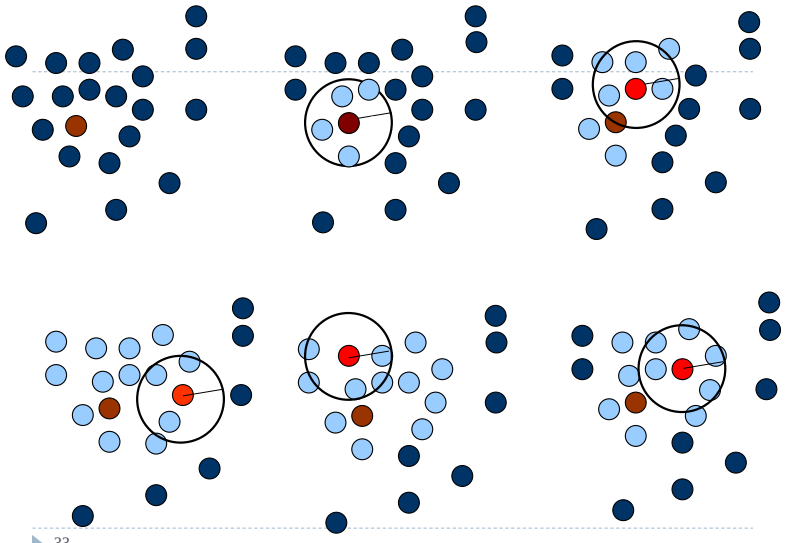
Contoh DBSCAN



1. Periksa the ε-neighborhood dari p;
2. Jika p memiliki lebih sedikit tetangga dari MinPts maka tandai p sebagai pencilan/outlier dan lanjutkan dengan objek selanjutnya
3. Jika tidak, tandai p sebagai yang telah diproses dan letakkan semua tetangganya di kluster C

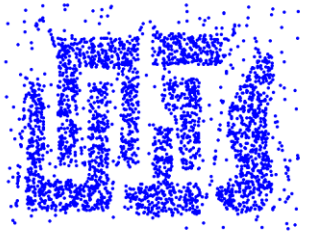
1. Periksa semua objek di C yang belum diproses
2. Jika bukan core object, return C
3. Jika tidak (jika core object), ambil secara acak satu core object p_1, tandai p_1 sebagai yang telah diproses (processed), dan letakkan semua unprocessed neighbors dari p_1 di kluster C

Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt



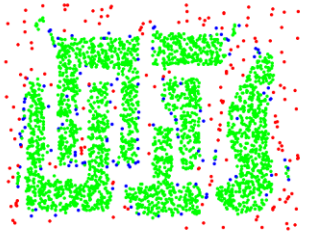
Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Contoh DBSCAN



Original Points

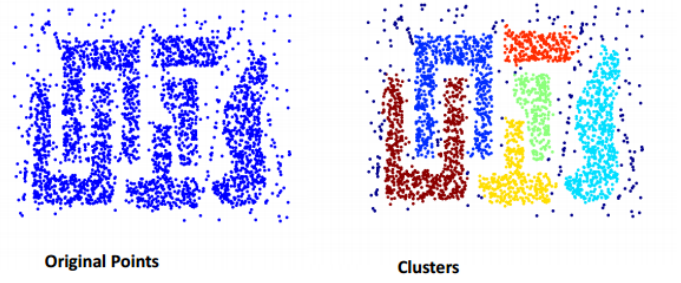
ε = 10, MinPts = 4



Point types: core, border and outliers

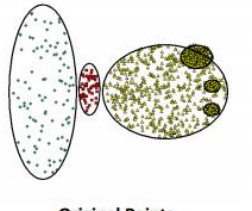
Source: www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

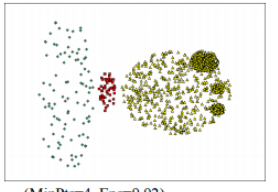
When DBSCAN Works Well

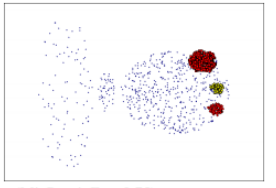


- Tahan terhadap Noise
- Dapat menangani kluster dari bentuk dan ukuran yang berbeda

When DBSCAN Does NOT Work Well




(MinPts=4, Eps=9.92).


(MinPts=4, Eps=9.75)

- Tidak mampu menangani berbagai variasi kepadatan
- Sensitif terhada parameter-sulit untuk menentukan himpunan parameter yang benar parameters

Reference

- Tan P., Michael S., & Vipin K. 2006. *Introduction to Data mining*. Pearson Education, Inc. – Chapter 10
- Han J & Kamber M. 2006. *Data mining – Concept and Techniques*. 2nd Edition, Morgan-Kauffman, San Diego - Chapter 7
- www.cse.buffalo.edu/faculty/azhang/cse601/density-based.ppt

Topik selanjutnya:
Teknik klasifikasi