



Data Mining

#10 Meeting

Anomaly Detection

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001





Pengertian Anomaly

Anomali (Outlier) adalah sebuah observasi data yang menyimpang secara signifikan dari sisa data lainnya. Anomali merupakan titik data yang "berbeda" atau "tidak biasa" dibandingkan dengan mayoritas data dalam dataset.

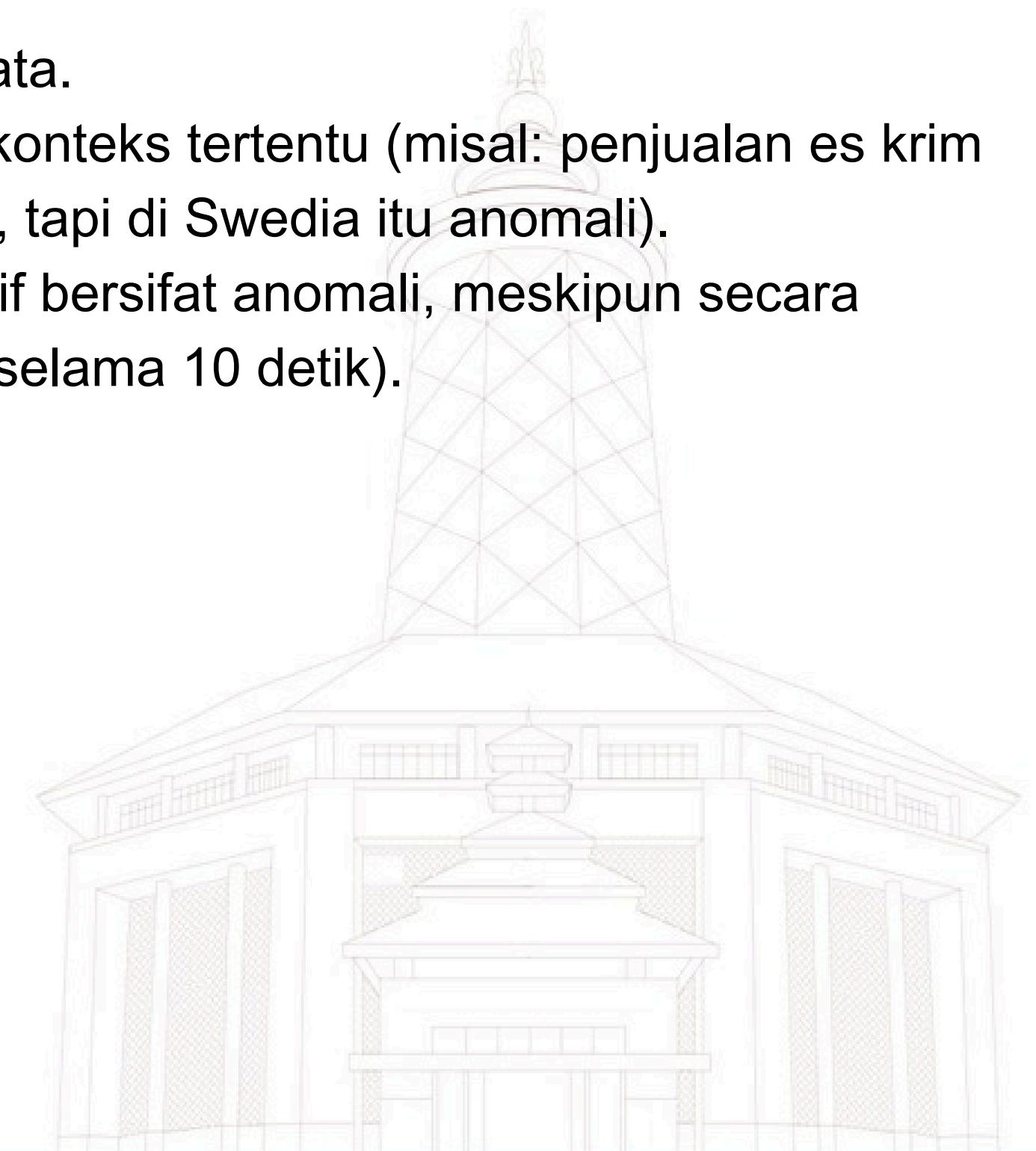
Mengapa Deteksi Anomali Penting?

- Pembersihan Data (Error): Seringkali anomali adalah hasil dari kesalahan input (misal: usia = 500 tahun) atau kegagalan sensor.
- Deteksi Intrusi/Fraud: Dalam keamanan siber atau transaksi keuangan, anomali seringkali menjadi sinyal adanya aktivitas ilegal (misal: pola transaksi kartu kredit yang tiba-tiba sangat berbeda).
- Penemuan Baru (Innovation): Dalam sains, anomali bisa jadi bukan error, melainkan sebuah penemuan baru atau fenomena langka yang menarik untuk diteliti.



Tipe Anomali

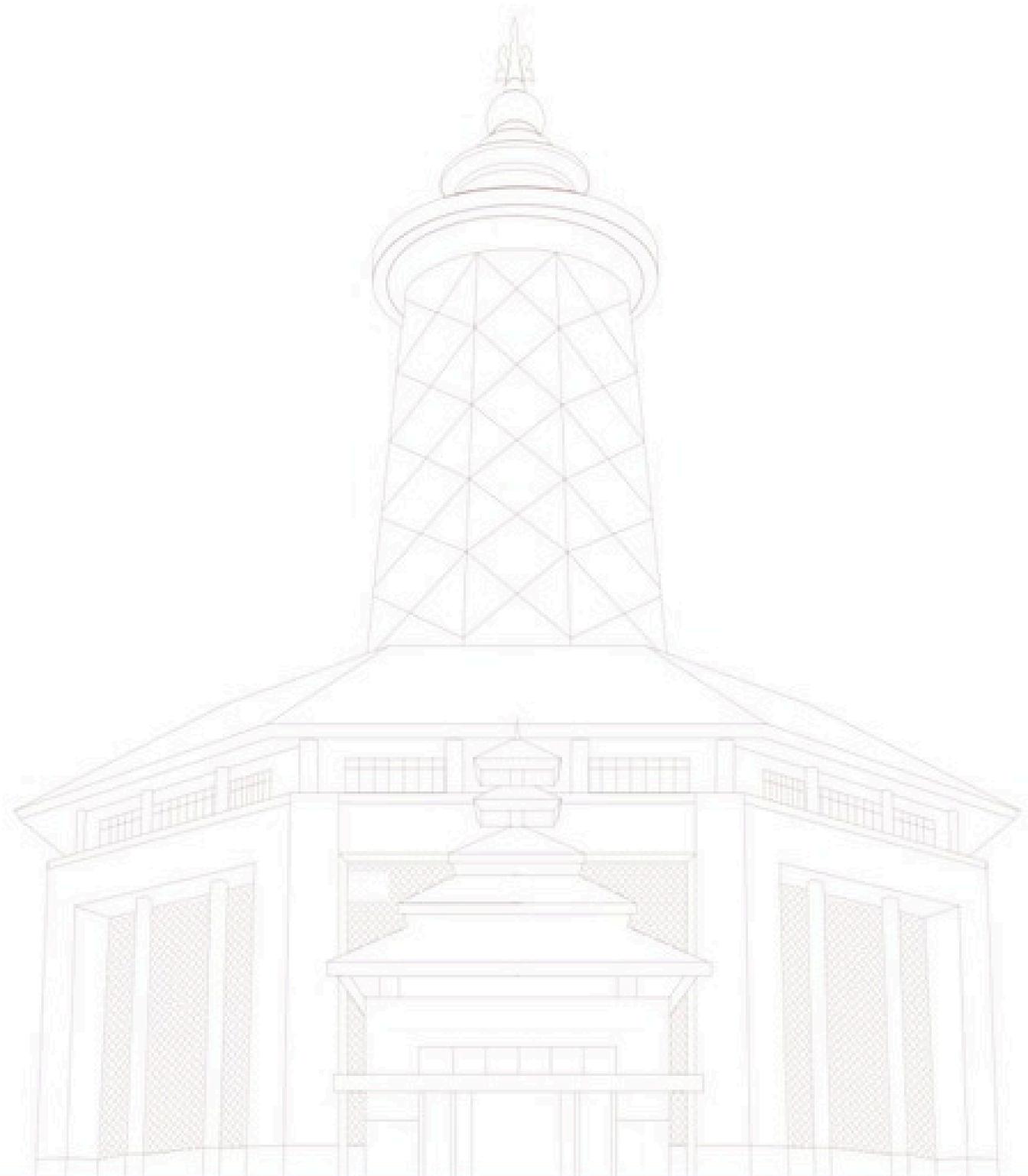
- Point Anomaly: Satu titik data individual yang jauh dari sisa data.
- Contextual Anomaly: Sebuah observasi yang anomali dalam konteks tertentu (misal: penjualan es krim sangat tinggi di bulan Desember. Di Indonesia mungkin wajar, tapi di Swedia itu anomali).
- Collective Anomaly: Sekumpulan titik data yang secara kolektif bersifat anomali, meskipun secara individual tidak (misal: detak jantung yang mendatar (flatline) selama 10 detik).





Deteksi Anomali

- Deteksi dengan pendekatan Statistik (Univariat)
- Deteksi Anomali Proximity-Based (Distance-Based)
- Deteksi Anomali Density-Based
- Deteksi Anomali Clustering-Based





Deteksi - Pendekatan Statistik

Ada dua metode utama :

- Metode Z-Score (berbasis Mean & Standar Deviasi)
- Metode IQR / Box Plot (berbasis Median & Kuartil)

Kita akan menggunakan satu set data yang sama untuk membandingkan keduanya:

data_usia = [25, 30, 22, 28, 35, 40, 29, 31, 27, 55, 33, 26, 1, 150]





Deteksi - Pendekatan Statistik

1. Metode Z-Score (Standard Score)

Metode ini mengukur seberapa jauh sebuah titik data dari rata-rata (mean), dalam satuan standar deviasi.

- Asumsi: Data harus terdistribusi normal (simetris, bentuk lonceng).
- Kelemahan: Sangat tidak robust. Nilai mean dan standar deviasi itu sendiri sangat sensitif (mudah "tertarik") oleh outlier.
- Aturan Umum: Sebuah data dianggap anomali jika nilai absolut Z-Score-nya lebih besar dari 3 (atau 2.5, atau 2, tergantung sensitivitas). Kita akan pakai $|Z| > 2$ untuk contoh ini.



Deteksi - Pendekatan Statistik

Data: [25, 30, 22, 28, 35, 40, 29, 31, 27, 55, 33, 26, 1, 150] ($N=14$), threshold $|Z| > 2$ dianggap anomali,
Deteksi dengan Z-Score

Formula: $Z = \frac{(x-\mu)}{\sigma}$

Langkah 1: Hitung Rata-rata (Mean, μ)

- $\mu = (25 + 30 + \dots + 1 + 150)/14$
- $\mu = 472/14 = \mathbf{33.71}$

Langkah 2: Hitung Standar Deviasi (Sample, σ)

- $\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N-1}}$
- $\sum(x_i - \mu)^2 = (25 - 33.71)^2 + (30 - 33.71)^2 + \dots + (1 - 33.71)^2 + (150 - 33.71)^2$
- $\sum(x_i - \mu)^2 \approx 15480.29$
- $\sigma = \sqrt{15480.29/(14 - 1)} = \sqrt{1190.79} \approx \mathbf{34.51}$

Langkah 3: Hitung Z-Score untuk Setiap Data (Contoh)

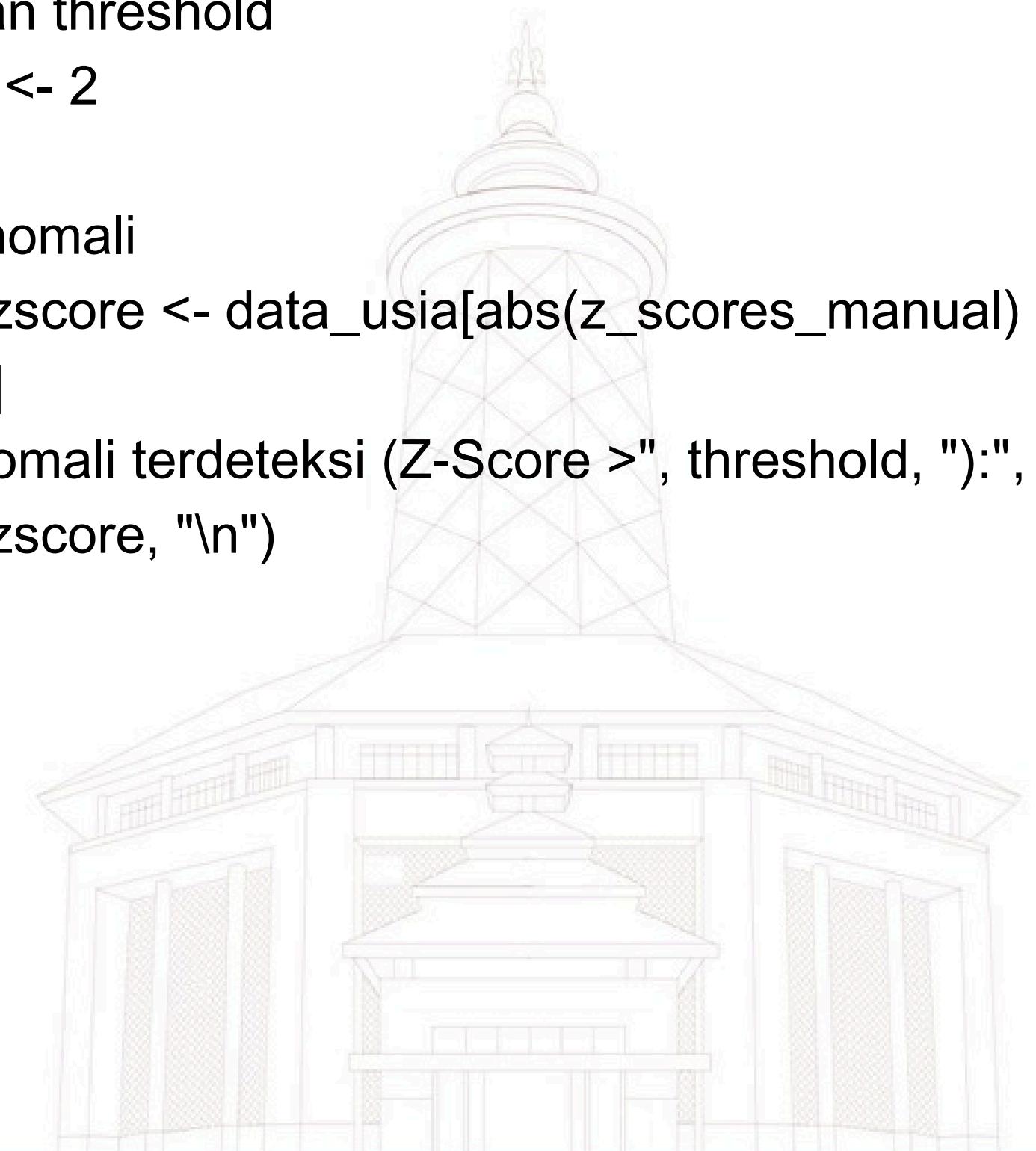
- Z-Score (data normal, 25): $Z = (25 - 33.71)/34.51 = -8.71/34.51 = \mathbf{-0.25}$
(Normal)
- Z-Score (anomali, 150): $Z = (150 - 33.71)/34.51 = 116.29/34.51 = \mathbf{+3.37}$
(Anomali)
- Z-Score (anomali, 1): $Z = (1 - 33.71)/34.51 = -32.71/34.51 = \mathbf{-0.95}$ (Dianggap Normal!)



Deteksi - Pendekatan Statistik

```
# Data dummy  
data_usia <- c(25, 30, 22, 28, 35, 40, 29, 31, 27, 55,  
33, 26, 1, 150)  
  
# Cara 1: Fungsi scale() (cara cepat)  
z_scores_auto <- scale(data_usia)  
print(z_scores_auto)  
  
# Cara 2: Perhitungan manual (sesuai hitungan kita)  
mean_usia <- mean(data_usia)  
sd_usia <- sd(data_usia)  
z_scores_manual <- (data_usia - mean_usia) / sd_usia  
  
cat("\n--- Hasil Z-Score ---\n")  
print(data.frame(Usia = data_usia, ZScore =  
z_scores_manual))
```

```
# Tentukan threshold  
threshold <- 2  
  
# Filter anomali  
anomali_zscore <- data_usia[abs(z_scores_manual) >  
threshold]  
cat("\nAnomali terdeteksi (Z-Score >", threshold, "):",  
anomali_zscore, "\n")
```



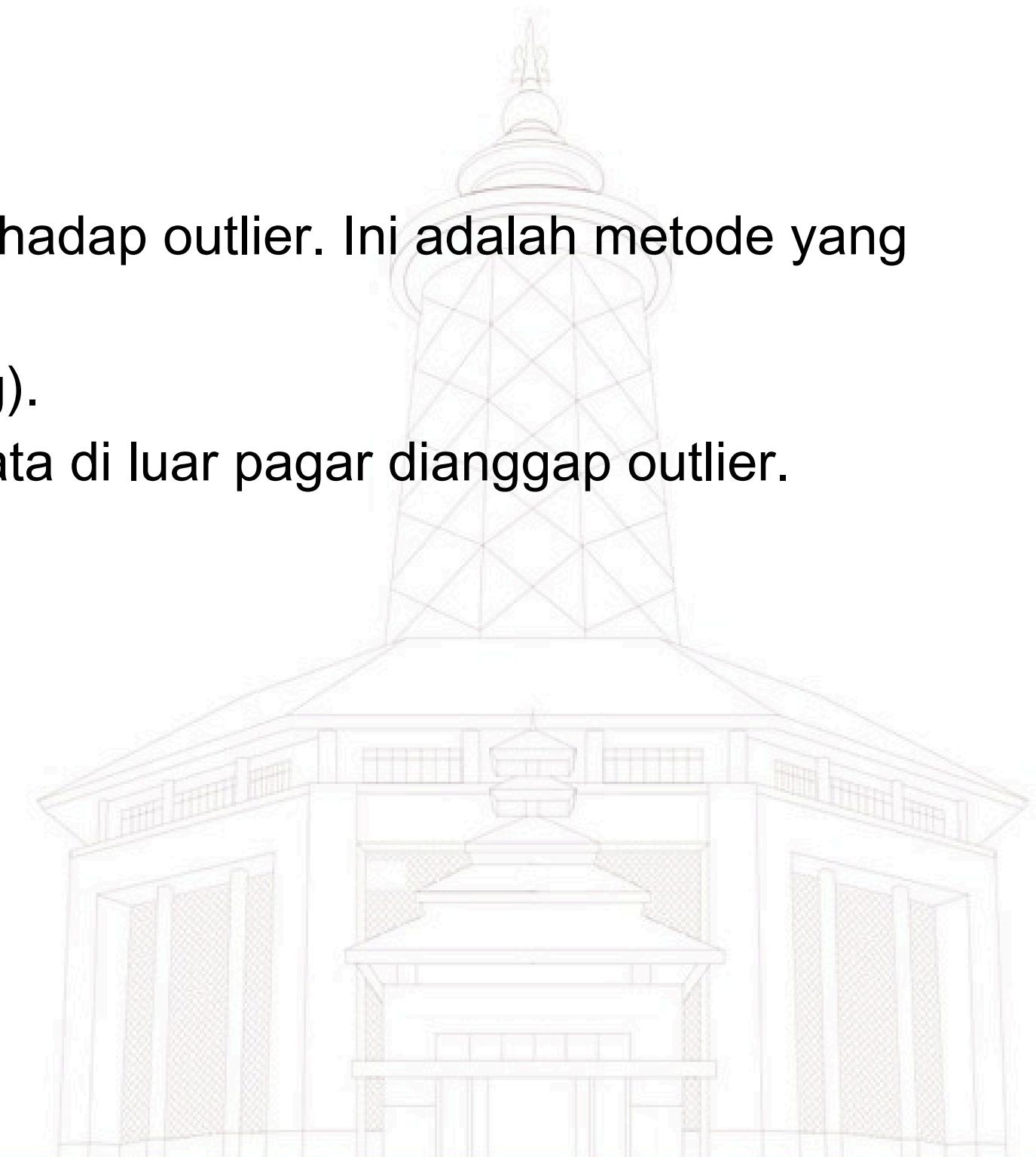


Deteksi - Pendekatan Statistik

2. Metode IQR (Interquartile Range) / Box Plot

Metode ini adalah metode non-parametrik yang robust (tahan) terhadap outlier. Ini adalah metode yang digunakan untuk menggambar Box Plot.

- Asumsi: Tidak ada. Cocok untuk data yang skewed (menceng).
- Logika: Data yang wajar berada di dalam "pagar" (fences). Data di luar pagar dianggap outlier.
- Aturan Umum: Data adalah anomali jika:
 - $x > (Q3 + 1.5 \times IQR)$ (Pagar Atas)
 - $x < (Q1 - 1.5 \times IQR)$ (Pagar Bawah)





Deteksi - Pendekatan Statistik

Data: [25, 30, 22, 28, 35, 40, 29, 31, 27, 55, 33, 26, 1, 150] (N=14).

Deteksi dengan IQR



Langkah 1: Urutkan Data (N=14) 1, 22, 25, 26, 27, 28, 29, 30, 31, 33, 35, 40, 55,
150

Langkah 2: Cari Median (Q2)

- Data tengah adalah rata-rata data ke-7 dan ke-8.
- Median (Q2) = $(29 + 30)/2 = 29.5$

Langkah 3: Cari Kuartil 1 (Q1)

- Median dari paruh bawah data: [1, 22, 25, 26, 27, 28, 29]
- Data tengah (data ke-4) adalah **26**.
- **Q1 = 26**

Langkah 4: Cari Kuartil 3 (Q3)

- Median dari paruh atas data: [30, 31, 33, 35, 40, 55, 150]
- Data tengah (data ke-4) adalah **35**.
- **Q3 = 35**



Deteksi - Pendekatan Statistik

Langkah 5: Hitung IQR

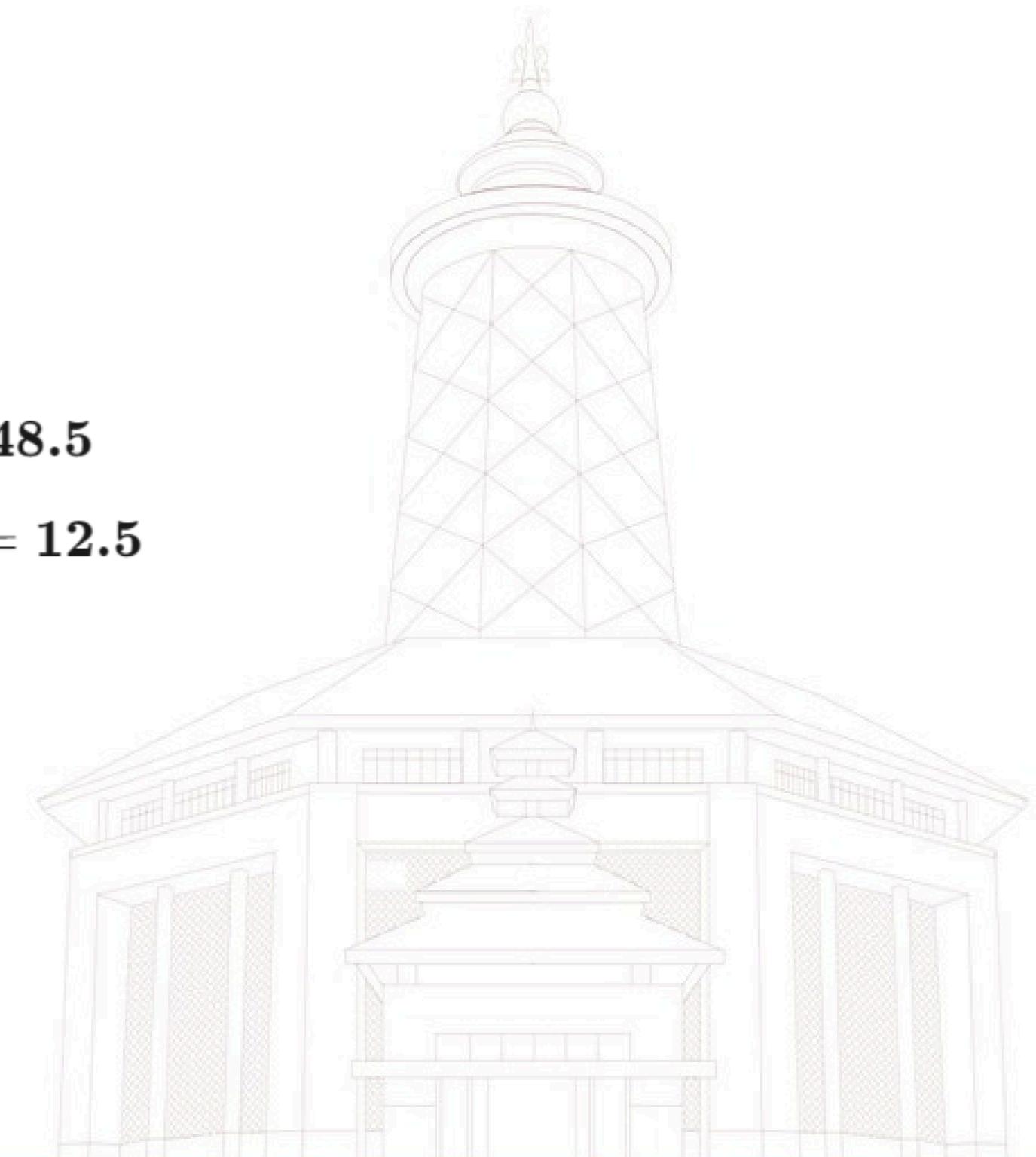
- $IQR = Q3 - Q1 = 35 - 26 = 9$

Langkah 6: Tentukan Pagar (Fences)

- **Pagar Atas:** $Q3 + (1.5 \times IQR) = 35 + (1.5 \times 9) = 35 + 13.5 = 48.5$
- **Pagar Bawah:** $Q1 - (1.5 \times IQR) = 26 - (1.5 \times 9) = 26 - 13.5 = 12.5$

Kesimpulan (IQR):

- Data 150 > 48.5 → **Anomali**
- Data 55 > 48.5 → **Anomali**
- Data 1 < 12.5 → **Anomali**





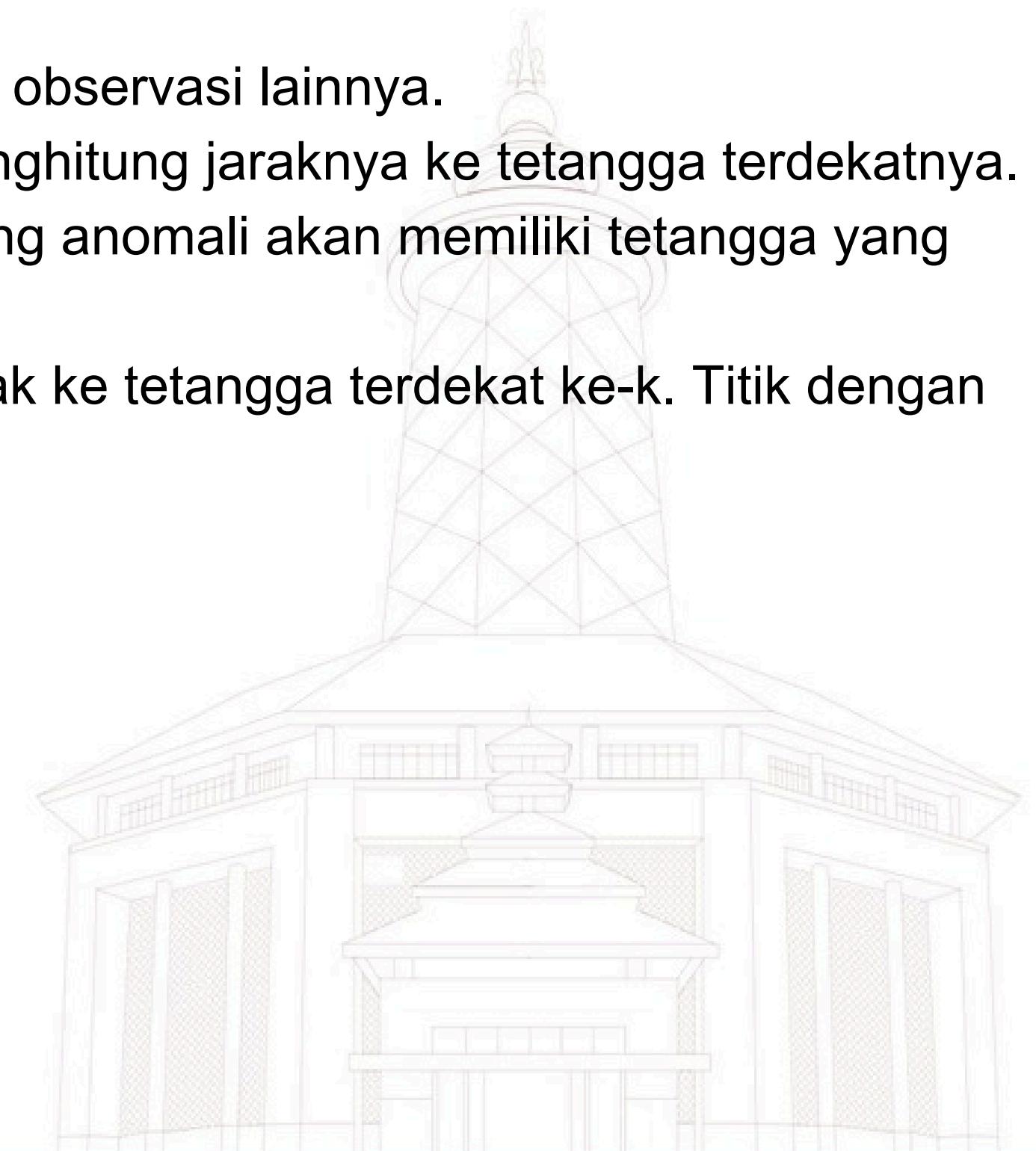
Deteksi - Pendekatan Statistik

```
# Data dummy  
data_usia <- c(25, 30, 22, 28, 35, 40, 29, 31, 27, 55,  
33, 26, 1, 150)  
  
# --- Visualisasi Langsung ---  
# Boxplot secara otomatis menghitung dan  
menampilkan outlier (titik-titik)  
boxplot(data_usia,  
        main="Boxplot Deteksi Anomali Usia",  
        ylab="Usia",  
        col="lightblue",  
        border="blue")  
mtext("Titik di luar 'kumis' adalah outlier (anomali)",  
side=1)  
  
# quantile() R mungkin sedikit berbeda dari hitungan  
manual  
# (ada 9 tipe metode!), tapi konsepnya sama.  
Q1 <- quantile(data_usia, 0.25)  
Q3 <- quantile(data_usia, 0.75)  
IQR_val <- IQR(data_usia) # Cara cepat: IQR()  
# Hitung pagar  
pagar_atas <- Q3 + (1.5 * IQR_val)  
pagar_bawah <- Q1 - (1.5 * IQR_val)  
cat("Pagar Bawah:", pagar_bawah, "\n")  
cat("Pagar Atas:", pagar_atas, "\n")  
# Filter anomali  
anomali_iqr <- data_usia[data_usia > pagar_atas |  
data_usia < pagar_bawah]  
cat("\nAnomali terdeteksi (IQR):", anomali_iqr, "\n")
```



Deteksi - Proximity-Based (Distance-Based)

- Konsep Utama: Anomali adalah observasi yang terisolasi dari observasi lainnya.
- Logika: Kita mengukur keterisolasi sebuah titik dengan menghitung jaraknya ke tetangga terdekatnya. Titik yang normal akan memiliki tetangga yang dekat. Titik yang anomali akan memiliki tetangga yang sangat jauh.
- Metrik: Skor Jarak k-NN (k-Nearest Neighbor). Kita hitung jarak ke tetangga terdekat ke-k. Titik dengan skor (jarak) terbesar adalah anomali.





Deteksi - Proximity-Based (Distance-Based)

Data: [25, 30, 22, 28, 35, 40, 29, 31, 27, 55, 33, 26, 1, 150] (N=14).

Deteksi dengan Distance Based (k-NN / k- Nearest Neighboor)

Parameter: Kita pilih $k = 2$. (Kita akan mencari jarak ke tetangga terdekat ke-2).

Langkah 1: Hitung Jarak ke Tetangga ke-2 untuk SEMUA Titik Kita akan hitung jarak absolut

$|x_i - x_j|$ untuk setiap titik.

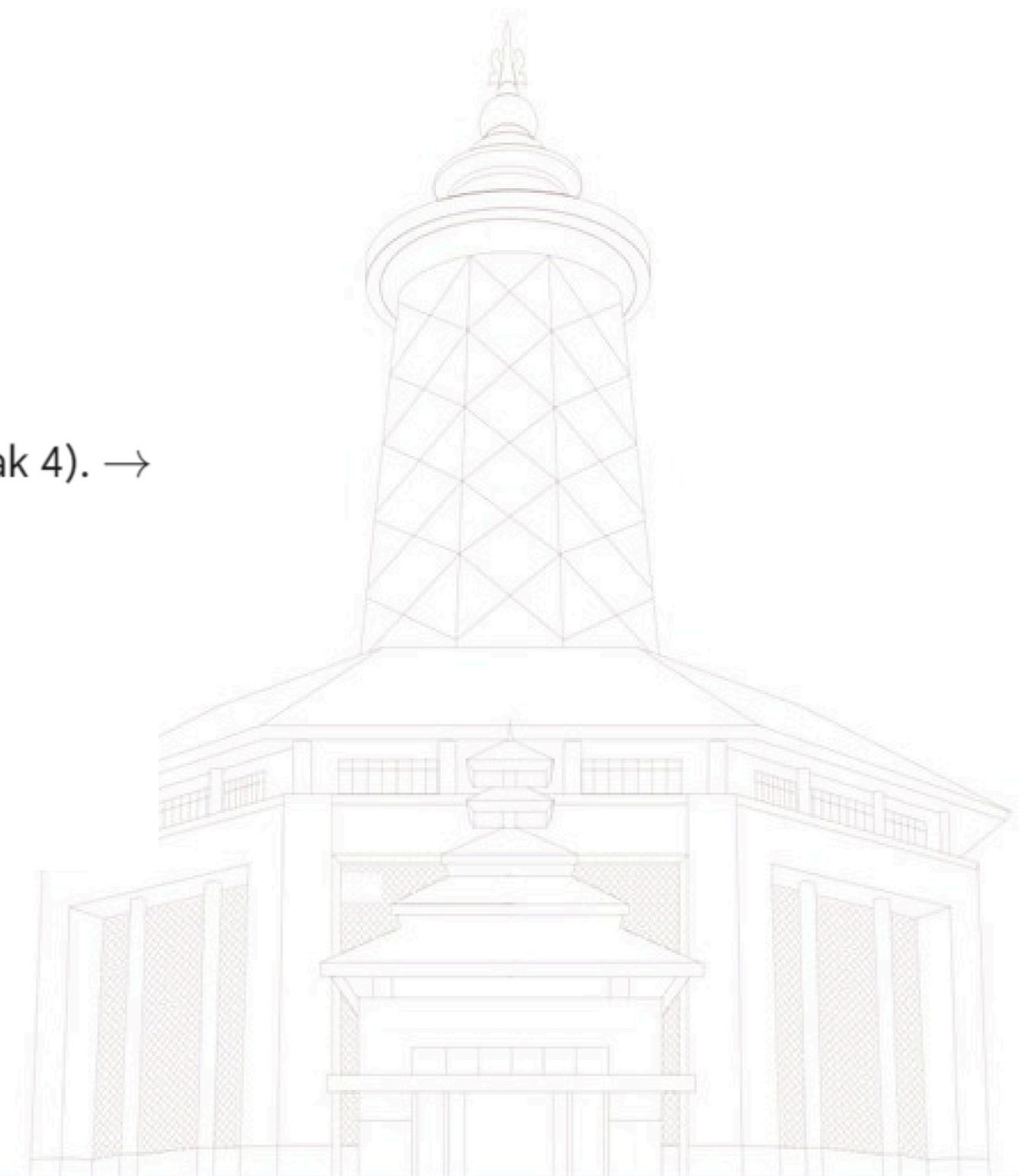
1. **Titik 1** : Tetangga: 22 (jarak 21), 25 (jarak 24). \rightarrow **Skor (k=2) = 24**
2. **Titik 22** : Tetangga: 25 (jarak 3), 26 (jarak 4). \rightarrow **Skor (k=2) = 4**
3. **Titik 25** : Tetangga: 26 (jarak 1), 27 (jarak 2). \rightarrow **Skor (k=2) = 2**
4. **Titik 26** : Tetangga: 25 (jarak 1), 27 (jarak 1). \rightarrow **Skor (k=2) = 1**
5. **Titik 27** : Tetangga: 26 (jarak 1), 28 (jarak 1). \rightarrow **Skor (k=2) = 1**
6. **Titik 28** : Tetangga: 27 (jarak 1), 29 (jarak 1). \rightarrow **Skor (k=2) = 1**
7. **Titik 29** : Tetangga: 28 (jarak 1), 30 (jarak 1). \rightarrow **Skor (k=2) = 1**





Deteksi - Proximity-Based (Distance-Based)

8. **Titik 30** : Tetangga: 29 (jarak 1), 31 (jarak 1). → **Skor (k=2) = 1**
9. **Titik 31** : Tetangga: 30 (jarak 1), 33 (jarak 2). → **Skor (k=2) = 2**
10. **Titik 33** : Tetangga: 31 (jarak 2), 35 (jarak 2). → **Skor (k=2) = 2**
11. **Titik 35** : Tetangga: 33 (jarak 2), 40 (jarak 5). Koreksi: 33 (jarak 2), 31 (jarak 4). →
Skor (k=2) = 4
12. **Titik 40** : Tetangga: 35 (jarak 5), 33 (jarak 7). → **Skor (k=2) = 7**
13. **Titik 55** : Tetangga: 40 (jarak 15), 35 (jarak 20). → **Skor (k=2) = 20**
14. **Titik 150** : Tetangga: 55 (jarak 95), 40 (jarak 110). → **Skor (k=2) = 110**





Deteksi - Proximity-Based (Distance-Based)

Peringkat	Titik Data (Usia)	Skor Anomali (Jarak k=2)	Keterangan
1	150	110	(Sangat Jauh)
2	1	24	(Jauh)
3	55	20	(Jauh)
---	(SIKU / KNEE)	--- LOMPATAN BESAR ---	(Cutoff di sini)
4	40	7	(Agak Jauh)
5	22	4	(Normal)
6	35	4	(Normal)
7	25	2	(Sangat Normal)
8	31	2	(Sangat Normal)
9	33	2	(Sangat Normal)
10	26	1	(Klaster Padat)
11	27	1	(Klaster Padat)
12	28	1	(Klaster Padat)
13	29	1	(Klaster Padat)
14	30	1	(Klaster Padat)





Deteksi - Proximity-Based (Distance-Based)

1. Install dan load library

```
library(dbSCAN)
```

2. Data kita

```
data_usia <- c(25, 30, 22, 28, 35, 40, 29, 31, 27, 55,  
33, 26, 1, 150)
```

```
k <- 2
```

3. Hitung skor anomali (jarak ke k=2)

Ubah ke matrix 1-kolom agar bisa dibaca kNNdist

```
data_matrix <- as.matrix(data_usia)
```

```
skor_anomali <- kNNdist(data_matrix, k = k)
```

4. Urutkan skor dari tertinggi ke terendah

```
skor_terurut <- sort(skor_anomali, decreasing = TRUE)
```

5. Tampilkan skor terurut di console

(Ini akan mencetak: 110.0, 24.0, 20.0, 7.0, 4.0, ...)

```
print("Skor Anomali (Jarak k=2) Diurutkan:")
```

```
print(skor_terurut)
```

6. Buat 1 Plot (Knee Plot)

Ini adalah satu-satunya plot yang kita butuhkan
plot(skor_terurut,

```
type = "b", # "b" = both (titik dan garis)
```

```
main = "Knee Plot (Plot Siku) Skor Anomali",
```

```
ylab = "Skor Anomali (Jarak)",
```

```
xlab = "Peringkat Titik Data",
```

```
pch = 19, # Simbol bulat
```

```
col = "blue")
```

Tambahkan garis penanda "siku" (knee) secara sederhana

Kita menandai 'setelah' peringkat 3

```
abline(v = 3.5, lty = 2, col = "red")
```

```
text(x = 2, y = 80, "Anomali", col = "red")
```

```
text(x = 8, y = 40, "Normal", col = "blue")
```



Deteksi - Proximity-Based (Distance-Based)

Data:

- A = (1, 1)
- B = (1, 2)
- C = (2, 1)
- O = (5, 5)

Deteksi dengan Distance Based (k-NN / k- Nearest Neighboor), k=2, dan jaraknya Euclidian

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Jarak antar titik :

- $d(A, B) = \sqrt{(1-1)^2 + (2-1)^2} = \sqrt{1} = 1.0$
- $d(A, C) = \sqrt{(2-1)^2 + (1-1)^2} = \sqrt{1} = 1.0$
- $d(A, O) = \sqrt{(5-1)^2 + (5-1)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$
- $d(B, C) = \sqrt{(2-1)^2 + (1-2)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$
- $d(B, O) = \sqrt{(5-1)^2 + (5-2)^2} = \sqrt{16+9} = \sqrt{25} = 5.0$
- $d(C, O) = \sqrt{(5-2)^2 + (5-1)^2} = \sqrt{9+16} = \sqrt{25} = 5.0$





Deteksi - Proximity-Based (Distance-Based)

- **Skor Titik A (1, 1):**

- Jarak terdekat: $d(A, B) = 1.0$ (1st NN) dan $d(A, C) = 1.0$ (2nd NN).
- Jarak ke tetangga ke-2 adalah 1.0.
- **Skor Anomali (A) = 1.0**

- **Skor Titik B (1, 2):**

- Jarak terdekat (urut): $d(B, A) = 1.0$ (1st NN), $d(B, C) \approx 1.41$ (2nd NN), $d(B, O) = 5.0$ (3rd NN).
- Jarak ke tetangga ke-2 adalah 1.41.
- **Skor Anomali (B) = 1.41**

- **Skor Titik C (2, 1):**

- Jarak terdekat (urut): $d(C, A) = 1.0$ (1st NN), $d(C, B) \approx 1.41$ (2nd NN), $d(C, O) = 5.0$ (3rd NN).
- Jarak ke tetangga ke-2 adalah 1.41.
- **Skor Anomali (C) = 1.41**

- **Skor Titik O (5, 5) (Anomali):**

- Jarak terdekat (urut): $d(O, B) = 5.0$ (1st NN), $d(O, C) = 5.0$ (2nd NN), $d(O, A) \approx 5.66$ (3rd NN).
- Jarak ke tetangga ke-2 adalah 5.0.
- **Skor Anomali (O) = 5.0**



Peringkat	Titik	Skor Anomali (Jarak k=2)	Keterangan
1	O		5 ANOMALI
---	---	--- LOMPATAN BESAR ---	(Cutoff di sini)
2	B		1.41 Normal
3	C		1.41 Normal
4	A		1 Normal



Deteksi - Density Based

Konsep Utama: Metode ini adalah evolusi dari proximity-based.

- Proximity-Based (k-NN) bertanya: "Seberapa jauh tetanggamu?"
- Density-Based (LOF) bertanya: "Seberapa padat wilayahmu dibandingkan dengan kepadatan wilayah tetanggamu?"

Analogi Sederhana:

- Titik A (Normal): Rumah di tengah klaster perumahan padat. Kepadatan Anda (A) = TINGGI. Kepadatan tetangga Anda = TINGGI. (Ratio ~ 1).
- Titik B (Anomali): Rumah Anda sendirian di tengah hutan. Kepadatan Anda (O) = SANGAT RENDAH. Tetangga terdekat Anda adalah rumah di "Titik A". Kepadatan tetangga Anda = TINGGI.



Deteksi - Density Based

Algoritma Utama: LOF (Local Outlier Factor)

- LOF menghitung skor yang pada dasarnya adalah rasio:

$$LOF \approx \frac{\text{Rata-rata Kepadatan Tetangga}}{\text{Kepadatan Titik Itu Sendiri}}$$

- Cara Membaca Skor LOF:
- $LOF \approx 1$: Normal. Kepadatan Anda sama dengan tetangga Anda (Anda bagian dari klaster).
- $LOF > 1$: Anomali. Kepadatan Anda jauh lebih rendah daripada tetangga Anda (Anda terisolasi). Semakin tinggi nilainya, semakin anomali.





Deteksi - Density Based

Data:

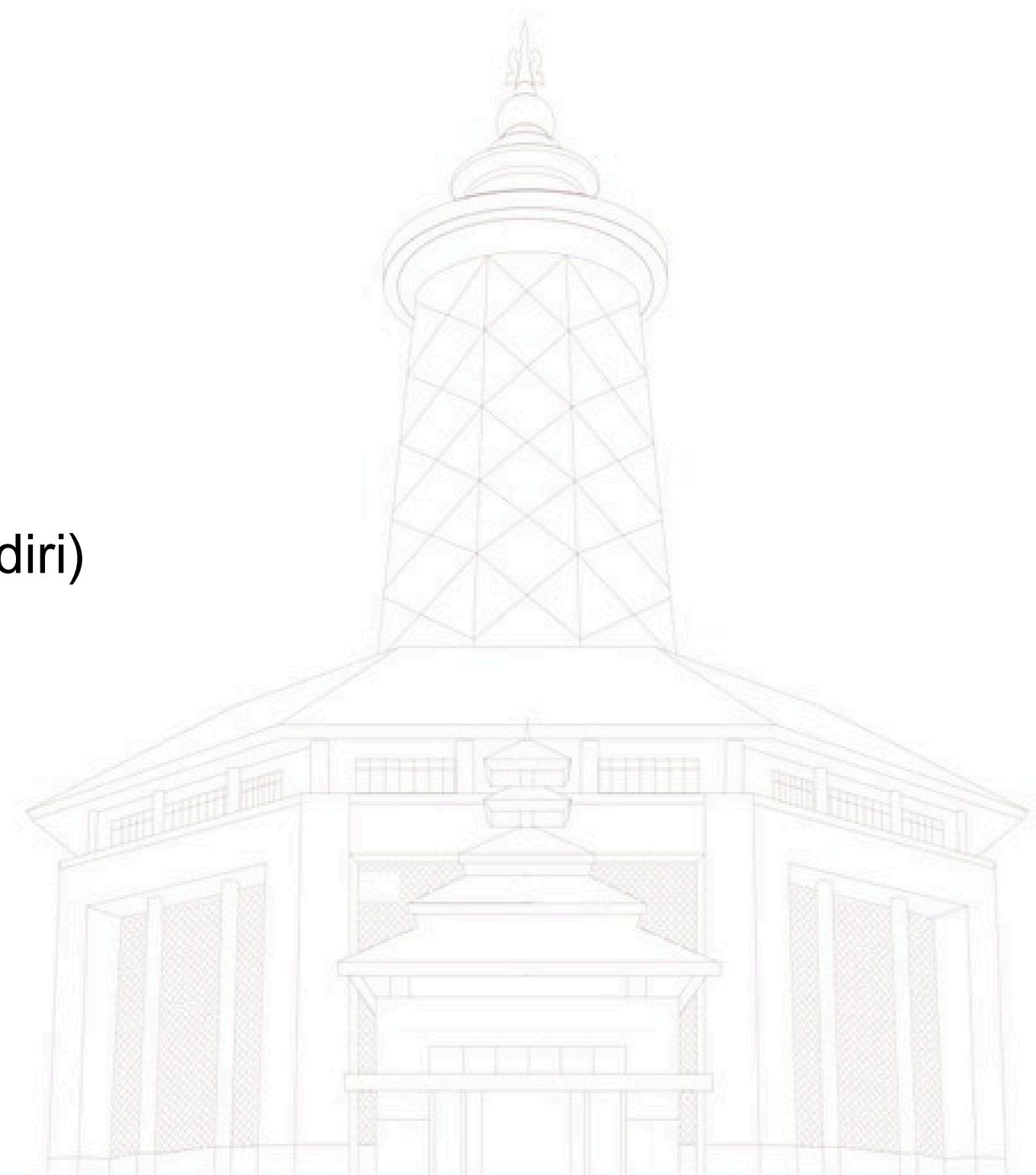
- $A = (1, 1)$
- $B = (1, 2)$
- $C = (2, 1)$
- $O = (5, 5)$

Deteksi dengan Density Based, $k=2$, jarak Euclidian

- Kepadatan Lokal (l_{rd}) = $1 / (\text{Rata-rata jarak ke } k=2 \text{ tetangga})$
- LOF = $(\text{Rata-rata Kepadatan Tetangga}) / (\text{Kepadatan Diri Sendiri})$

Jarak antar titik :

- $d(A, B) = \sqrt{(1-1)^2 + (2-1)^2} = \sqrt{1} = 1.0$
- $d(A, C) = \sqrt{(2-1)^2 + (1-1)^2} = \sqrt{1} = 1.0$
- $d(A, O) = \sqrt{(5-1)^2 + (5-1)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$
- $d(B, C) = \sqrt{(2-1)^2 + (1-2)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$
- $d(B, O) = \sqrt{(5-1)^2 + (5-2)^2} = \sqrt{16+9} = \sqrt{25} = 5.0$
- $d(C, O) = \sqrt{(5-2)^2 + (5-1)^2} = \sqrt{9+16} = \sqrt{25} = 5.0$





Deteksi - Density Based

Hitung Ird nya dahulu

- **Ird(A):**

- k=2 tetangga terdekat: **B** (jarak 1.0) dan **C** (jarak 1.0).
- Rata-rata jarak = $(1.0 + 1.0) / 2 = 1.0$
- **Ird(A) = 1 / 1.0 = 1.0**

- **Ird(B):**

- k=2 tetangga terdekat: **A** (jarak 1.0) dan **C** (jarak 1.41).
- Rata-rata jarak = $(1.0 + 1.41) / 2 = 1.205$
- **Ird(B) = 1 / 1.205 ≈ 0.83**

- **Ird(C):**

- k=2 tetangga terdekat: **A** (jarak 1.0) dan **B** (jarak 1.41).
- Rata-rata jarak = $(1.0 + 1.41) / 2 = 1.205$
- **Ird(C) = 1 / 1.205 ≈ 0.83**

- **Ird(O) (Anomali):**

- k=2 tetangga terdekat: **B** (jarak 5.0) dan **C** (jarak 5.0).
- Rata-rata jarak = $(5.0 + 5.0) / 2 = 5.0$
- **Ird(O) = 1 / 5.0 = 0.2**



Deteksi - Density Based

Kemudian hitung LOF

- **LOF(A):**

- Tetangga (k=2): B dan C.
- Rata-rata Ird tetangga = $(\text{Ird}(B) + \text{Ird}(C)) / 2 = (0.83 + 0.83) / 2 = 0.83$
- $\text{Ird}(A) = 1.0$
- **Skor LOF(A) = $0.83 / 1.0 = 0.83$** (Skor $\approx 1 \rightarrow$ NORMAL)

- **LOF(B):**

- Tetangga (k=2): A dan C.
- Rata-rata Ird tetangga = $(\text{Ird}(A) + \text{Ird}(C)) / 2 = (1.0 + 0.83) / 2 = 0.915$
- $\text{Ird}(B) = 0.83$
- **Skor LOF(B) = $0.915 / 0.83 = 1.1$** (Skor $\approx 1 \rightarrow$ NORMAL)

- **LOF(C):** (Sama dengan B)

- **Skor LOF(C) = 1.1** (Skor $\approx 1 \rightarrow$ NORMAL)

- **LOF(O) (Anomali):**

- Tetangga (k=2): B dan C.
- Rata-rata Ird tetangga = $(\text{Ird}(B) + \text{Ird}(C)) / 2 = (0.83 + 0.83) / 2 = 0.83$
- $\text{Ird}(O) = 0.2$
- **Skor LOF(O) = $0.83 / 0.2 = 4.15$** (Skor JAUH $> 1 \rightarrow$ ANOMALI)



Deteksi - Density Based

Kemudian hitung LOF

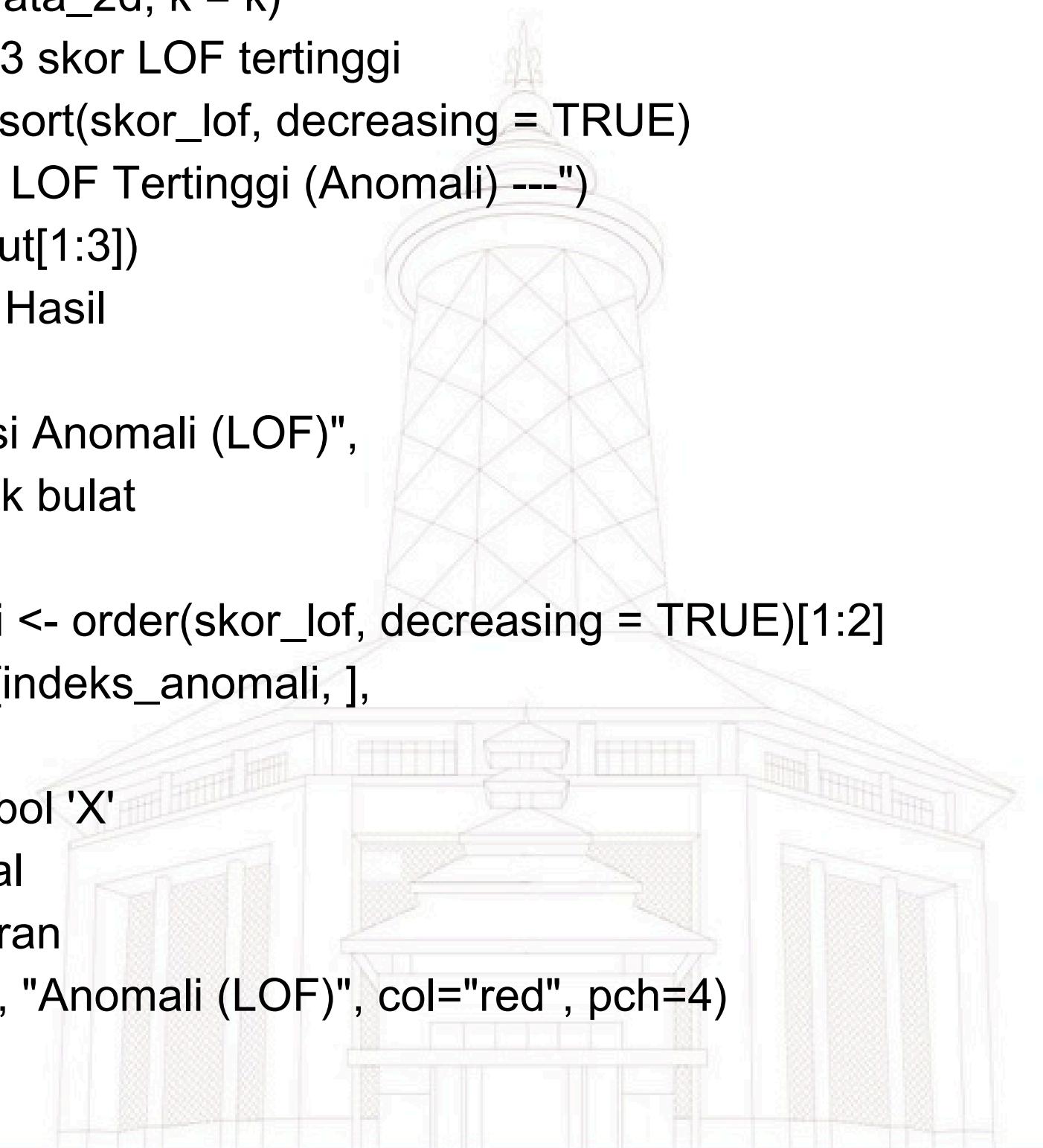
Titik	Kepadatan (lrd)	Rata-rata Kepadatan Tetangga	Skor LOF (Rasio)	Keterangan
A	1	0.83	0.83	Normal
B	0.83	0.915	1.1	Normal
C	0.83	0.915	1.1	Normal
O	0.2 (Rendah)	0.83 (Tinggi)	4.15	ANOMALI



Deteksi - Density Based

```
# 1. Install dan load library  
# install.packages("dbSCAN")  
library(dbSCAN)  
  
# 2. Buat data 2D  
set.seed(123) # Agar hasilnya selalu sama  
# Klaster Normal (50 titik, padat)  
klaster_normal <- data.frame(  
  x = rnorm(50, mean = 2, sd = 0.5),  
  y = rnorm(50, mean = 2, sd = 0.5)  
)  
  
# Anomali (2 titik buatan, terisolasi)  
anomali <- data.frame(  
  x = c(8, 1),  
  y = c(8, 6)  
)  
  
data_2d <- rbind(klaster_normal, anomali)  
  
# 3. Hitung skor anomali LOF  
# Kita pakai k=5 (nilai umum yang baik)  
k <- 5
```

```
skor_lof <- lof(data_2d, k = k)  
# 4. Tampilkan 3 skor LOF tertinggi  
skor_terurut <- sort(skor_lof, decreasing = TRUE)  
print("--- 3 Skor LOF Tertinggi (Anomali) ---")  
print(skor_terurut[1:3])  
  
# 5. Visualisasi Hasil  
plot(data_2d,  
  main = "Deteksi Anomali (LOF)",  
  pch = 19, # Titik bulat  
  col = "blue")  
indeks_anomali <- order(skor_lof, decreasing = TRUE)[1:2]  
points(data_2d[indeks_anomali, ],  
  col = "red",  
  pch = 4, # Simbol 'X'  
  lwd = 3, # Tebal  
  cex = 2) # Ukuran  
legend("topleft", "Anomali (LOF)", col="red", pch=4)
```

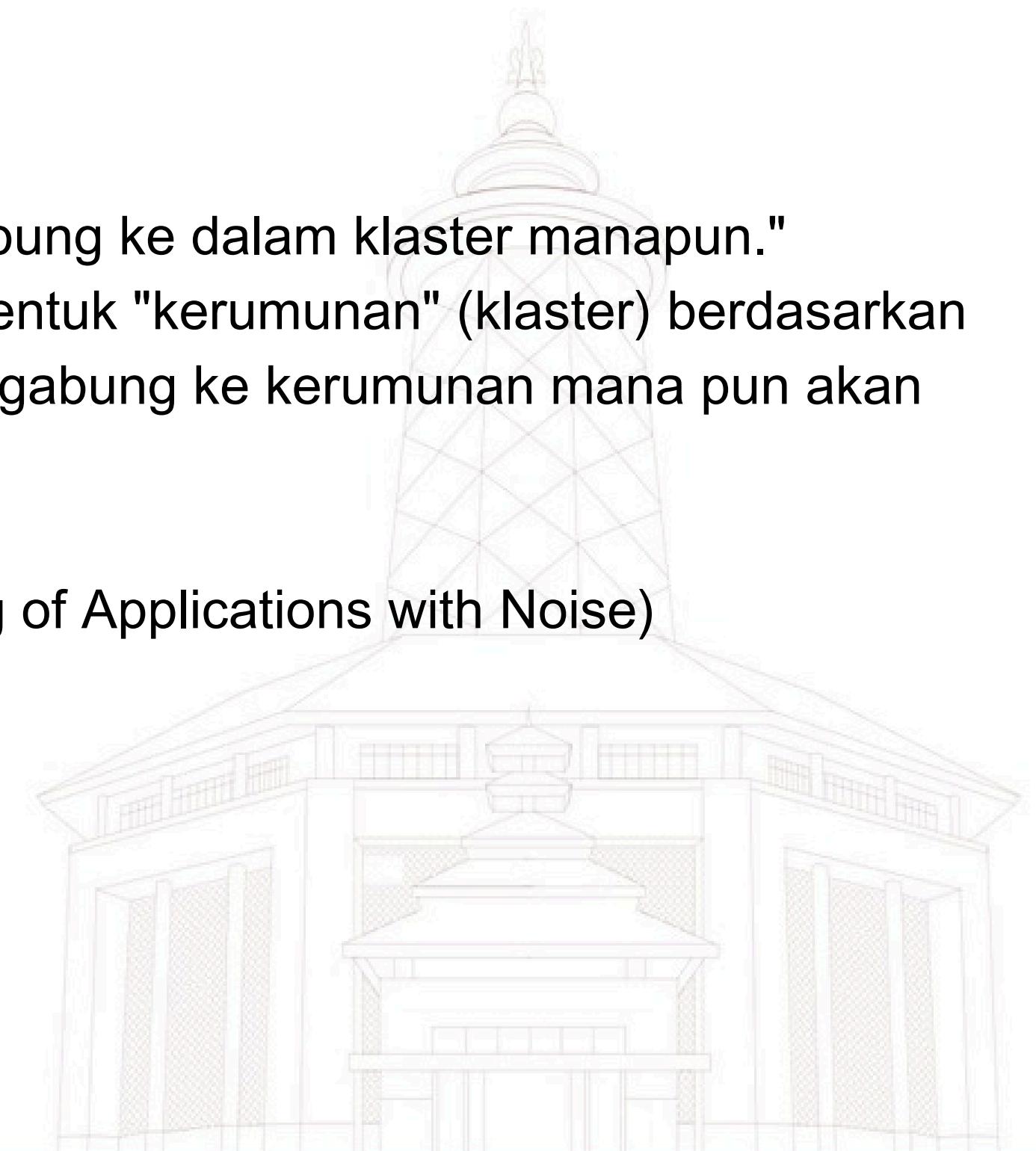




Deteksi - Clustering Based

Deteksi Anomali Clustering-Based (DBSCAN)

- Konsep Utama: "Anomali adalah observasi yang gagal bergabung ke dalam klaster manapun."
- Logika: Metode ini tidak menghitung skor. Ia mencoba membentuk "kerumunan" (klaster) berdasarkan kepadatan. Setiap titik data yang "tersesat" dan tidak bisa bergabung ke kerumunan mana pun akan dilabeli sebagai "Noise" (Gangguan).
- "Noise" inilah yang kita sebut sebagai ANOMALI.
- Algoritma Utama: DBSCAN (Density-Based Spatial Clustering of Applications with Noise)





Deteksi - Clustering Based

DBSCAN membutuhkan 2 parameter:

1. eps : Radius jarak. "Seberapa jauh jangkauan 'lingkaran' tetangga?"
2. minPts: Jumlah minimum titik. "Berapa banyak minimal tetangga di dalam lingkaran agar bisa disebut 'padat'?"

Berdasarkan 2 parameter itu, DBSCAN membagi titik menjadi 3 jenis:

- Core Point: Titik yang 'padat' (punya \geq minPts tetangga di dalam radius eps).
- Border Point: Titik yang 'tidak padat', tapi merupakan tetangga dari Core Point.
- Noise Point (ANOMALI): Titik yang bukan Core dan bukan Border. Dia sendirian.



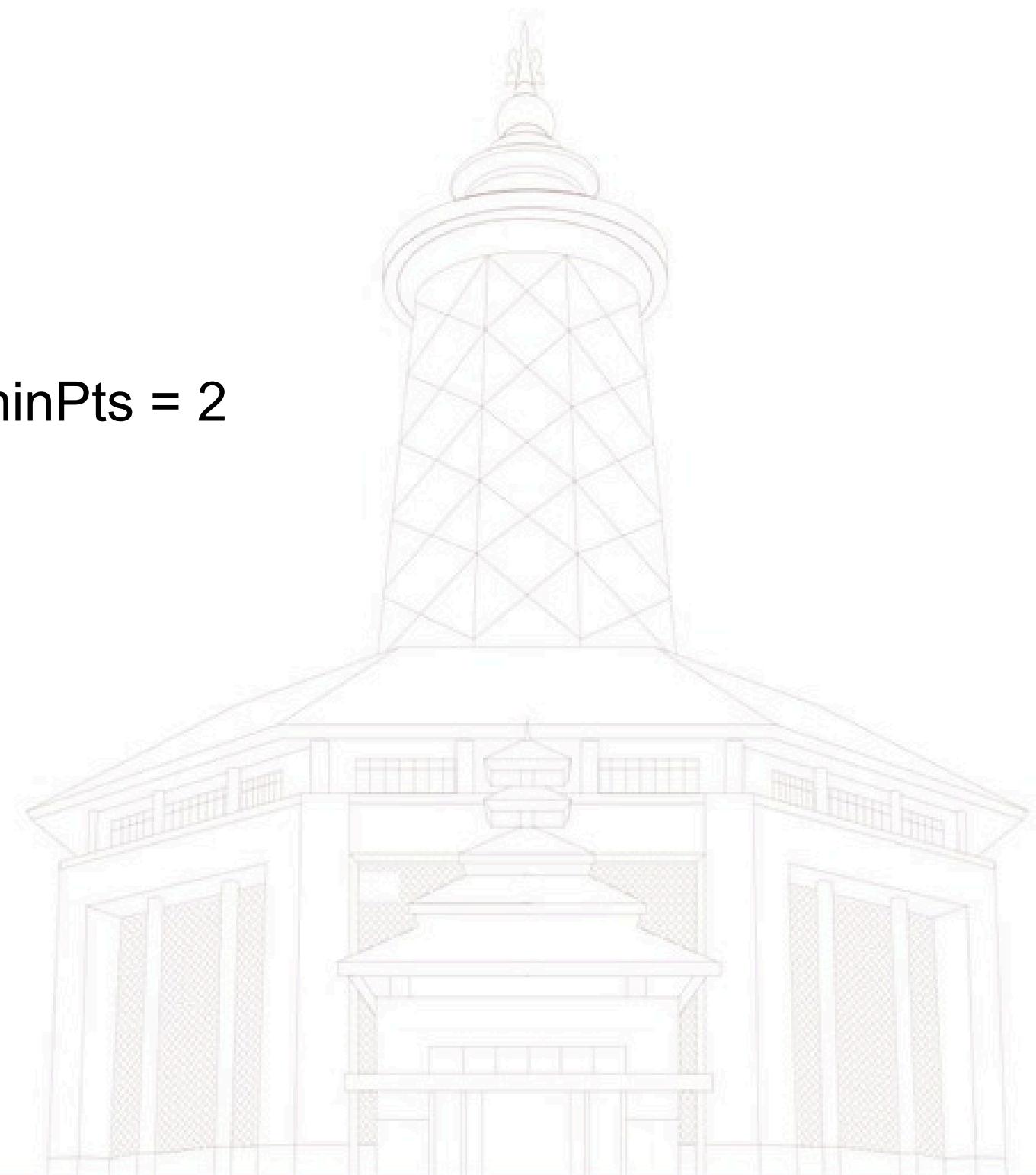
Deteksi - Clustering Based

Data:

- $A = (1, 1)$
- $B = (1, 2)$
- $C = (2, 1)$
- $O = (5, 5)$
- Deteksi dengan Clustering Based (DBSCAN), $\text{eps} = 1.5$ dan $\text{minPts} = 2$

Jarak antar titik :

- $d(A, B) = \sqrt{(1-1)^2 + (2-1)^2} = \sqrt{1} = 1.0$
- $d(A, C) = \sqrt{(2-1)^2 + (1-1)^2} = \sqrt{1} = 1.0$
- $d(A, O) = \sqrt{(5-1)^2 + (5-1)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$
- $d(B, C) = \sqrt{(2-1)^2 + (1-2)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$
- $d(B, O) = \sqrt{(5-1)^2 + (5-2)^2} = \sqrt{16+9} = \sqrt{25} = 5.0$
- $d(C, O) = \sqrt{(5-2)^2 + (5-1)^2} = \sqrt{9+16} = \sqrt{25} = 5.0$



Deteksi - Clustering Based

- **Analisis Titik A (1, 1):**
 - Tetangga (<1.5): **B** (1.0) dan **C** (1.0).
 - Jumlah tetangga = 2.
 - Apakah $2 \geq minPts (2)$? Ya. → A adalah **CORE Point**.
- **Analisis Titik B (1, 2):**
 - Tetangga (<1.5): **A** (1.0) dan **C** (1.41).
 - Jumlah tetangga = 2.
 - Apakah $2 \geq minPts (2)$? Ya. → B adalah **CORE Point**.
- **Analisis Titik C (2, 1):**
 - Tetangga (<1.5): **A** (1.0) dan **B** (1.41).
 - Jumlah tetangga = 2.
 - Apakah $2 \geq minPts (2)$? Ya. → C adalah **CORE Point**.
- **Analisis Titik O (5, 5):**
 - Tetangga (<1.5): Tidak ada.
 - Jumlah tetangga = 0.
 - Apakah $0 \geq minPts (2)$? Tidak. → O **bukan Core Point**.
 - Apakah O tetangga dengan Core Point? Tidak (jarak terdekatnya 5.0, $> eps$).
 - **Kesimpulan: O adalah NOISE Point (ANOMALI)**.



Deteksi - Clustering Based

```
# 1. Install dan load library  
# install.packages("dbSCAN")  
library(dbSCAN)  
  
# 2. Buat data 2D (SAMA PERSIS DENGAN CONTOH LOF)  
set.seed(123) # -> Diperlukan agar data 'random' (rnorm) selalu  
sama  
  
klaster_normal <- data.frame(  
  x = rnorm(50, mean = 2, sd = 0.5),  
  y = rnorm(50, mean = 2, sd = 0.5)  
)  
  
anomali <- data.frame(  
  x = c(8, 1),  
  y = c(8, 6)  
)  
  
data_2d <- rbind(klaster_normal, anomali)  
  
# 3. Jalankan DBSCAN  
# eps = radius lingkaran, minPts = minimum poin dalam  
lingkaran  
db_hasil <- dbSCAN(data_2d, eps = 0.8, minPts = 5)
```

```
# 4. Tampilkan hasil di console (Klaster 0 = Anomali)
```

```
print("Hasil Pembagian Klaster (0 = Anomali):")  
print(table(db_hasil$cluster))
```

```
# 5. Visualisasi Hasil
```

```
plot(data_2d,  
  main = "Deteksi Anomali (DBSCAN)",  
  pch = 19,  
  col = db_hasil$cluster + 1)
```

```
# Pindahkan legenda ke KANAN BAWAH (area ini kosong)
```

```
legend("bottomright", # <-- GANTI DARI "topleft"  
  legend = c("Anomali (Noise/Klaster 0)", "Klaster Normal  
(Klaster 1)",  
  col = c(1, 2), # 1=Hitam, 2=Merah  
  pch = 19)
```



SEE YOU NEXT WEEK !

Ferdian Bangkit Wijaya, S.Stat., M.Si

NIP. 199005202024061001

ferdian.bangkit@untirta.ac.id

