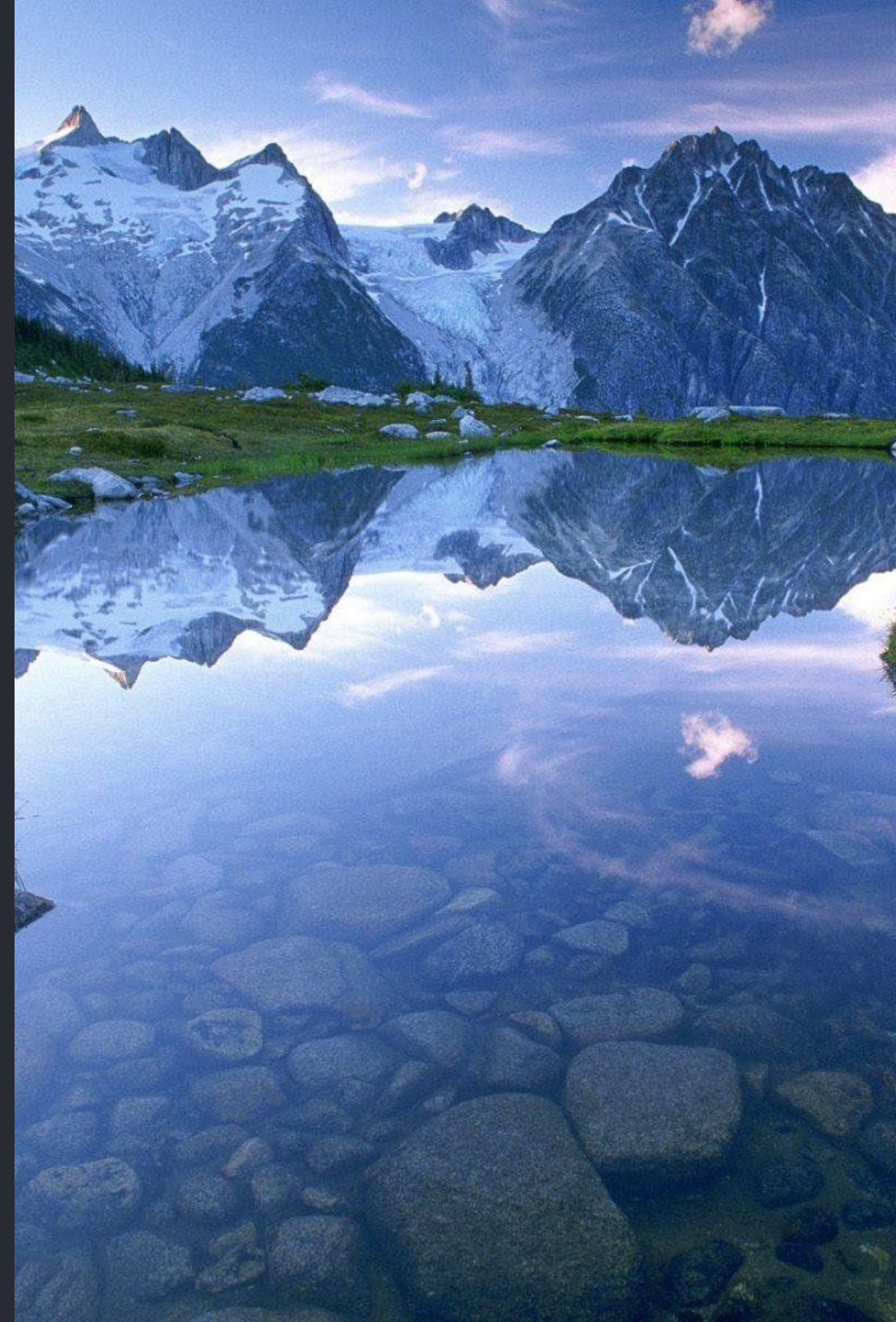


Klasterisasi Time Series

KELOMPOK 4

Reyhan Dela Masyhuri	20081010182
Renaldy William Lijaya Therry.	20081010179
Sri Fuji Santoso	20081010184
Linggar Bhakti Pratama	20081010185



Klasterisasi Time Series

Sebuah panduan tentang algoritma-algoritma klasterisasi (partisi, hierarki, berbasis grid, berbasis model, berbasis kepadatan, multi-step).





Klasterisasi Time Series dengan R

Menjelaskan proses klasterisasi time series dengan bahasa pemrograman R.

Partition-Based Clustering

K-Means

Memisahkan dataset menjadi K kelompok berdasarkan pusat terdekat.

Langkah utama: Inisialisasi pusat kluster, hitung jarak, kelompokkan, perbarui pusat.

Rentan terhadap outlier.

Cocok untuk data numerik.

K-Medoids

Memisahkan dataset menjadi K kelompok berdasarkan medoid terdekat.

Langkah utama: Inisialisasi medoid, hitung jarak, kelompokkan, perbarui medoid.

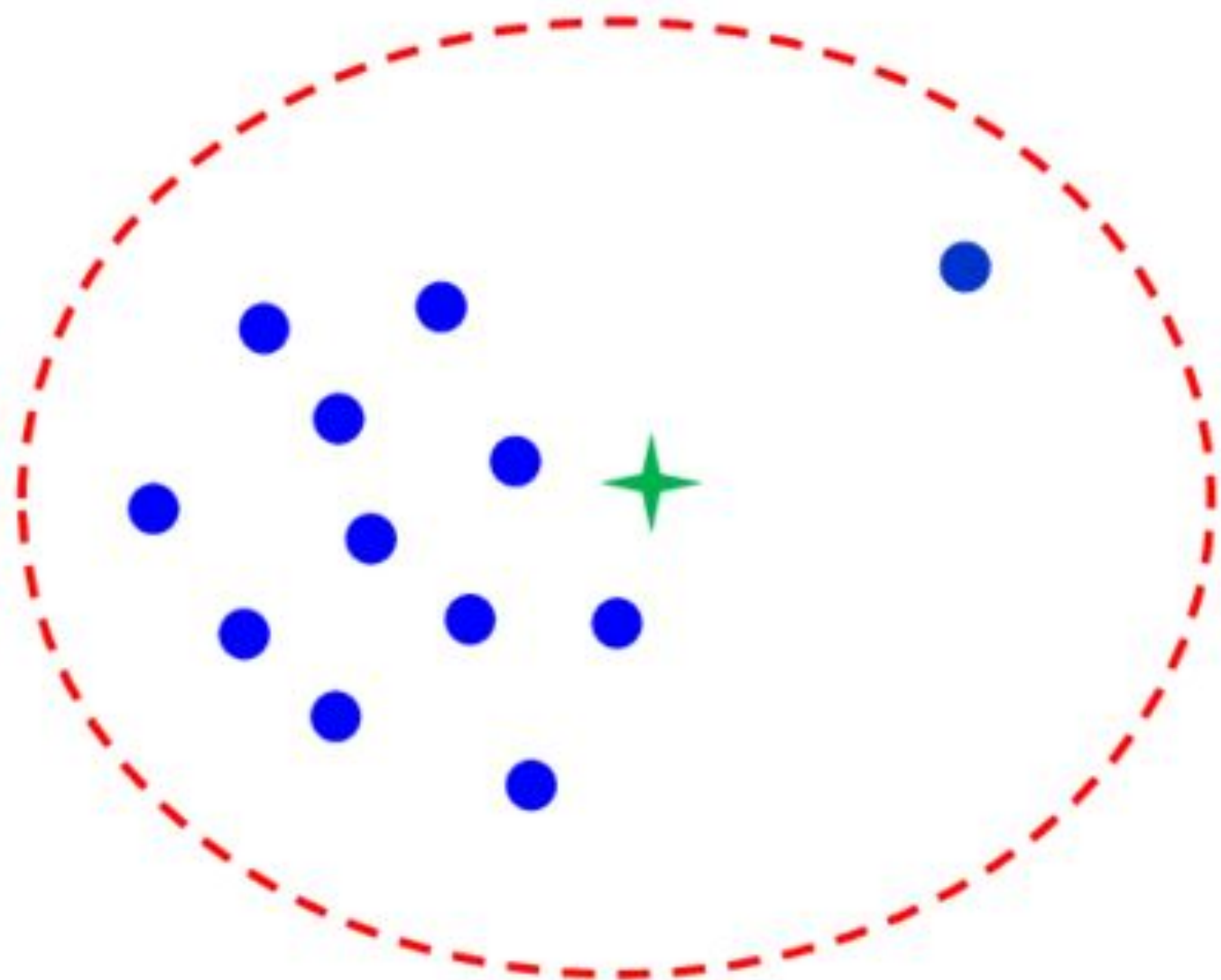
Lebih tahan terhadap outlier dan data non-numerik.

Cocok untuk data kategori/ordinal.

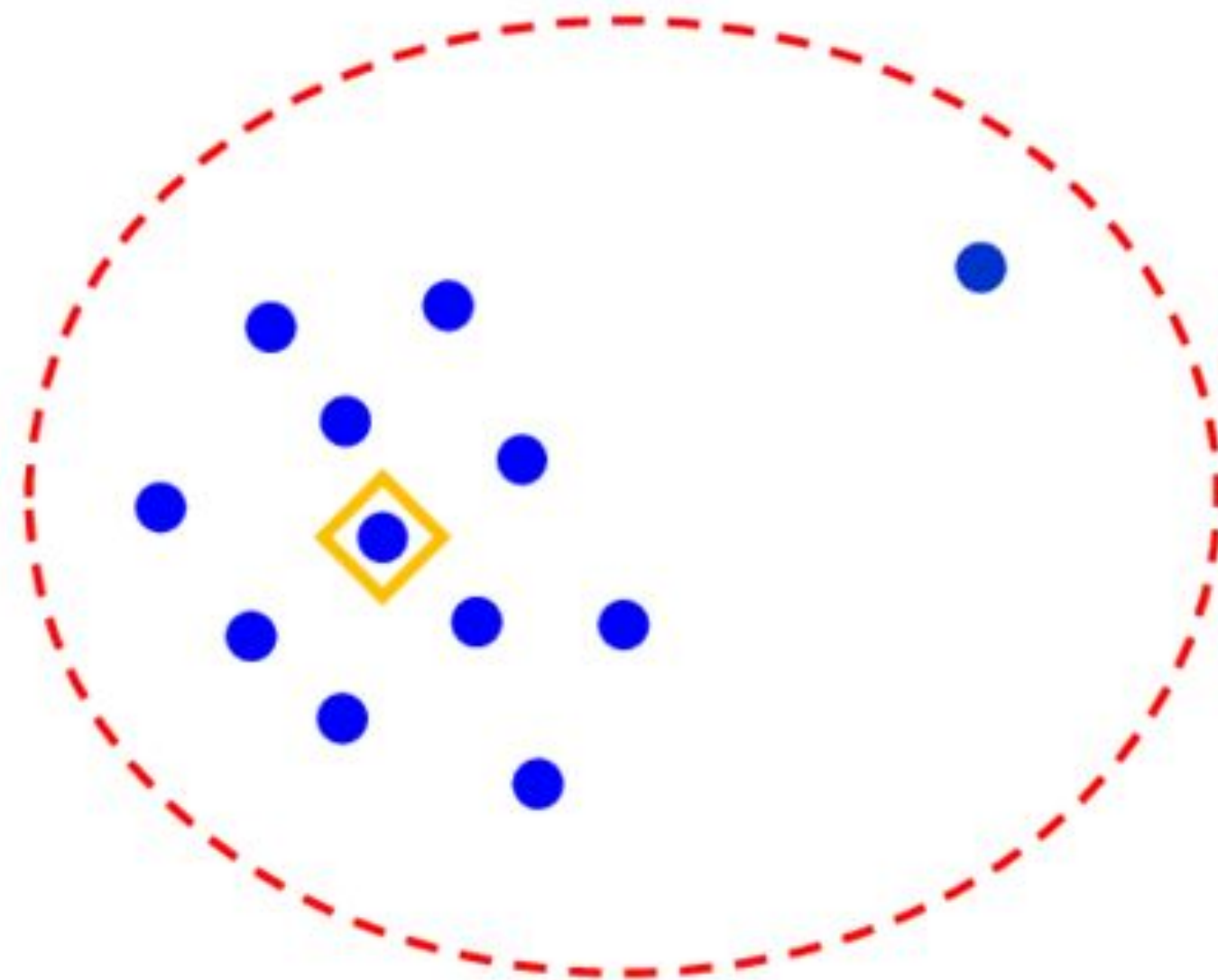
● Data Point

✦ Cluster Mean

◊ Cluster Medoid



K-Means Clustering



K-Medoids Clustering

K-Means

Adapun langkah-langkah untuk K-Means Clustering adalah sebagai berikut

(Prasetyo, 2014):

1. Inisialisasi: tentukan nilai k sebagai jumlah klaster yang diinginkan dan matriks jarak yang diinginkan.
2. Pilih k data dari set data X sebagai centroid.
3. Alokasikan semua data ke centroid terdekat dengan matriks jarak yang sudah ditetapkan (memperbarui klaster ID pada setiap data).
4. Hitung kembali centroid berdasarkan data yang mengikuti klaster masing-masing. Setiap pusat cluster dihitung ulang berdasarkan dari nilai rata-rata dalam cluster yang didapatkan
5. Ulangi langkah 3 dan 4 hingga kondisi konvergen tercapai, yaitu tidak ada data yang berpindah klaster.

K-Medoids

Adapun tahapan KMedoids Clustering adalah (Han dan Kamber, 2006).

1. Secara acak pilih k objek pada sekumpulan n objek sebagai medoid.
2. Ulangi.
3. Tempatkan objek non medoid ke dalam klaster yang paling dekat dengan medoid.
4. Secara acak pilih Orandom (sebuah objek non medoid).
5. Hitung total cost, S, dari pertukaran medoid O_j dengan Orandom. Hitung total cost (S) dengan menghitung nilai total jarak baru – total jarak lama
6. Jika $S < 0$ maka tukar O_j dengan Orandom untuk membentuk sekumpulan k objek baru sebagai medoid.
7. Ulangi hingga tidak ada perubahan

Hierarchical Clustering

1 Agglomerative

Algoritma ini memulai dengan setiap data sebagai kluster individual dan kemudian menggabungkannya berdasarkan jarak yang terukur.

2 Divisive

Algoritma ini memulai dengan semua data sebagai satu kluster dan kemudian membaginya menjadi kluster yang lebih kecil berdasarkan jarak.

3 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

Algoritma klasterisasi yang menggunakan hierarki dan mengurangi data secara iteratif.

4 CURE (Clustering Using Representatives)

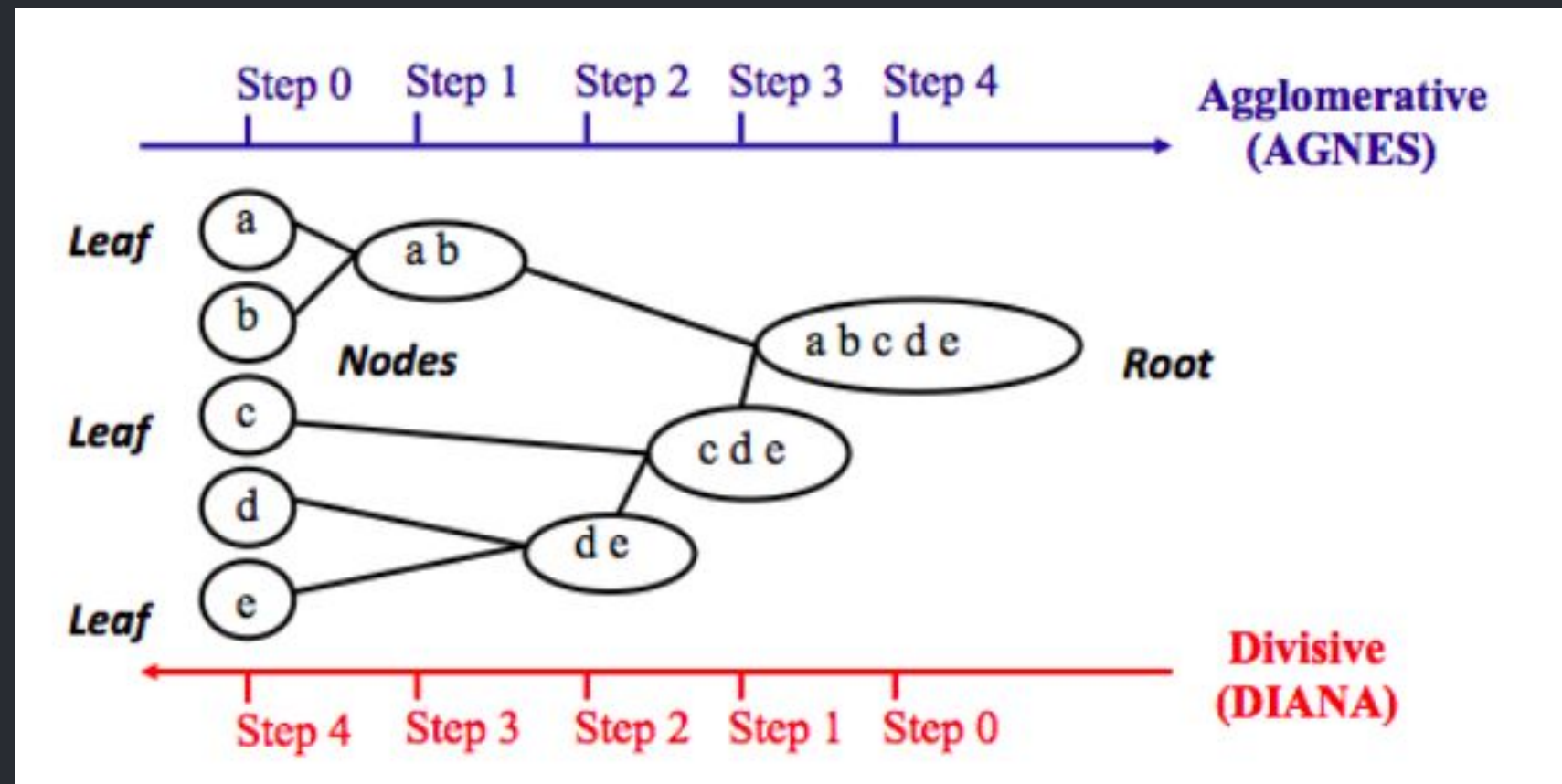
Algoritma klasterisasi yang merepresentasikan kluster dengan beberapa titik acuan.

Algoritma Agglomerative

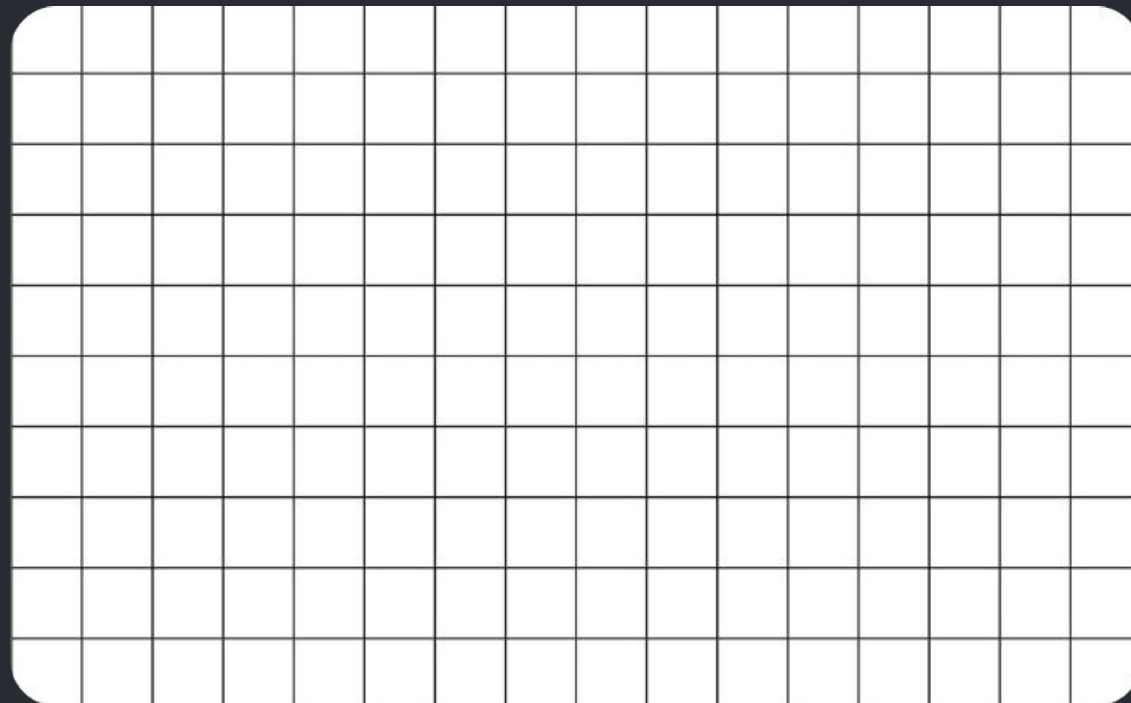
Algoritma untuk Agglomerative Clustering adalah sebagai berikut

1. Menyiapkan data, data yang digunakan bertipe numerik
2. Hitung jarak ukuran ketidaksamaan antar data (*dissimilarity measure*) , ini dapat dihitung menggunakan jarak *Euclidean* atau *Manhattan*. Nilai dari *dissimilarity measure* kemudian disusun menjadi *distance matriks* (matriks jarak)
3. Gabungkan dua cluster terdekat menggunakan pendekatan *linkage method* , beberapa metode *linkage* yang dapat digunakan adalah *complete linkage*, *single linkage* , *average linkage* , *centroid linkage*, dan *ward's minimum variance* , hasil perhitungan *linkage method* membentuk dendogram.
4. Menentukan di mana untuk memotong pohon hirarki (*hierarchical tree*) menjadi cluster, ini menciptakan partisi data
5. Melakukan analisis dari dendogram.

Perbedaan Klasterisasi Agglomerative dan Divisive



Grid-Based Clustering



STING (Statistical Information Grid)

Algoritma yang menggabungkan klasterisasi dengan grid berbasis statistik.



PreDe Con (Predictive Density-Based Clustering)

Algoritma klasterisasi berbasis density yang memprediksi kepadatan dalam suatu grid.



Model-Based Clustering

- EM (Expectation-Maximization): Algoritma yang memodelkan data menjadi kombinasi distribusi normal.

EM

Data berat badan (kg) : 50, 70, 43, 88, 96, 56

• Inisialisasi

- Kelompok "kurus" dengan rata-rata (μ_1) = 50 dan deviasi standar (σ_1) = 10.
- Kelompok "gemuk" dengan rata-rata (μ_2) = 80 dan deviasi standar (σ_2) = 10.

• Menghitung Probabilitas

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Dalam rumus ini:

- $f(x)$ adalah nilai PDF untuk variabel acak x .
- μ adalah rata-rata (mean) dari distribusi normal.
- σ adalah deviasi standar (standard deviation) dari distribusi normal.
- e adalah bilangan Euler, yaitu sekitar 2.71828.
- π adalah nilai pi, yaitu sekitar 3.14159.

Mari kita hitung $P(\text{"kurus"}|x)$ dan $P(\text{"gemuk"}|x)$ untuk $x = 70$:

Untuk "kurus":

$$P(\text{"kurus"}|70) = \frac{1}{10\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{70-50}{10}\right)^2} \approx 0.024$$

Untuk "gemuk":

$$P(\text{"gemuk"}|70) = \frac{1}{10\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{70-80}{10}\right)^2} \approx 0.035$$

- Perbaharui Parameter

Perbarui μ_1 dan σ_1 berdasarkan data yang termasuk dalam kelompok "kurus".

Misalkan kita memiliki data berat badan "kurus" sebagai berikut: [50, 43, 56]. Perbarui parameter "kurus" sebagai berikut:

- $\mu_1 = (50 + 43 + 56) / 3 = 49.67$
- $\sigma_1 = \sqrt{[(50 - 49.67)^2 + (43 - 49.67)^2 + (56 - 49.67)^2] / 3} = \sqrt{[78.33 / 3]} = \sqrt{26.11} \approx 5.11$

Perbarui μ_2 dan σ_2 berdasarkan data yang termasuk dalam kelompok "gemuk".

Misalkan kita memiliki data berat badan "gemuk" sebagai berikut: [70, 88, 96]. Perbarui parameter "gemuk" sebagai berikut:

- $\mu_2 = (70 + 88 + 96) / 3 = 84.67$
- $\sigma_2 = \sqrt{[(70 - 84.67)^2 + (88 - 84.67)^2 + (96 - 84.67)^2] / 3} = \sqrt{[334.33 / 3]} = \sqrt{111.44} \approx 10.56$

- Lakukan langkah-langkah Ekspektasi (Expectation) dan Maximisasi (Maximization) secara berulang sampai parameter stabil dan tidak berubah banyak antar iterasi.

Density-Based Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- DBSCAN mengidentifikasi klaster berdasarkan kerapatan data. Klaster didefinisikan sebagai wilayah di mana terdapat kerapatan pengamatan yang tinggi, dan daerah yang memiliki kerapatan yang lebih rendah dianggap sebagai noise.
- Tidak memerlukan jumlah klaster sebelumnya dan dapat menangani klaster dengan bentuk dan ukuran yang berbeda.

DBSCAN

Parameter-Parameter dalam DBSCAN

Epsilon (ϵ)

Jarak maksimum antar 2 titik agar memenuhi

MinPoints

jumlah minimum titik dalam suatu lingkungan agar suatu titik dianggap sebagai titik inti (ini termasuk titik itu sendiri)

Dengan menggunakan parameter sebelumnya, DBSCAN mengklasifikasikan titik-titik dataset menjadi :

Core Points

Titik A disebut titik inti jika setidaknya terdapat titik sebanyak minPoints (termasuk dirinya sendiri) dalam jarak ϵ darinya

Directly reachable points

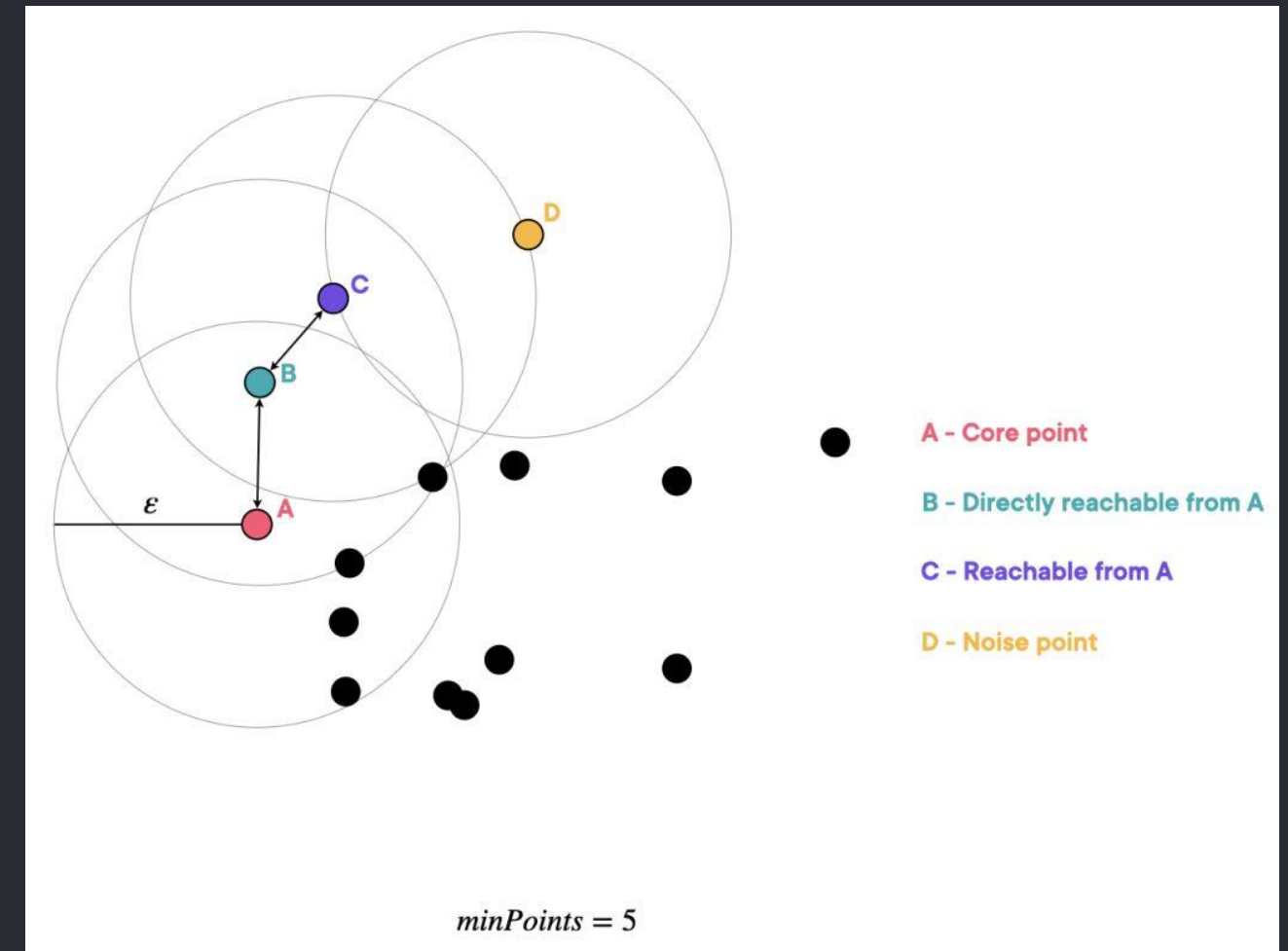
Titik B merupakan Directly reachable point dari A jika dapat dicapai secara langsung dari titik A (core point) dalam jarak ϵ dari titik A

Reachable points

Titik C dikatakan sebagai "reachable" dari titik A jika ada serangkaian titik yang membentuk jalur dari titik A ke titik C, dan semua titik dalam jalur tersebut adalah "directly reachable" dari titik A

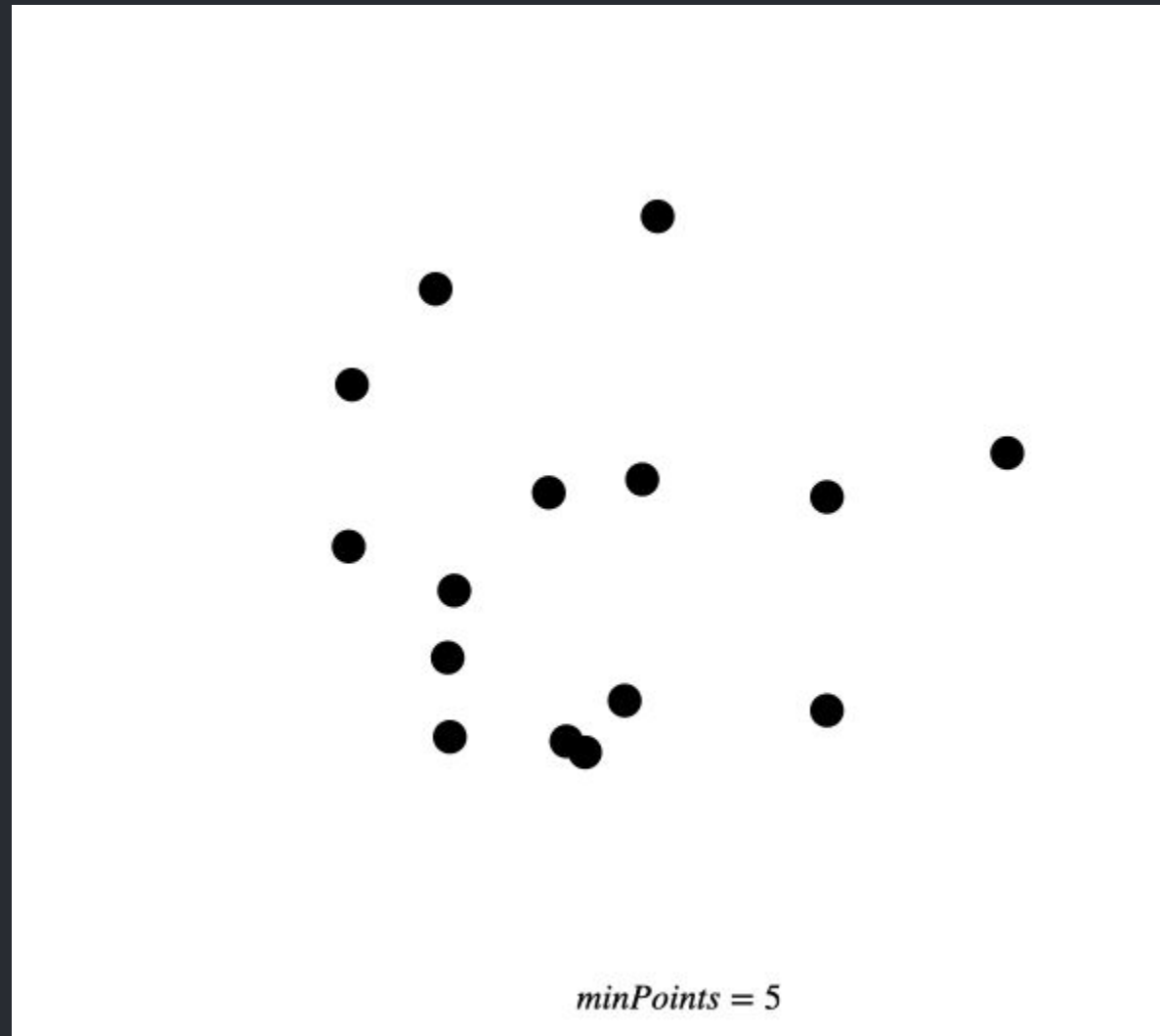
Noise points

jika sebuah titik tidak dapat dicapai dari titik lainnya dan jumlah titik yang ada di dalamnya di bawah MinPoints , maka titik tersebut dianggap sebagai outlier atau titik noise



Menentukan Cluster

Lingkaran-lingkaran yang saling bersinggungan merupakan 1 cluster

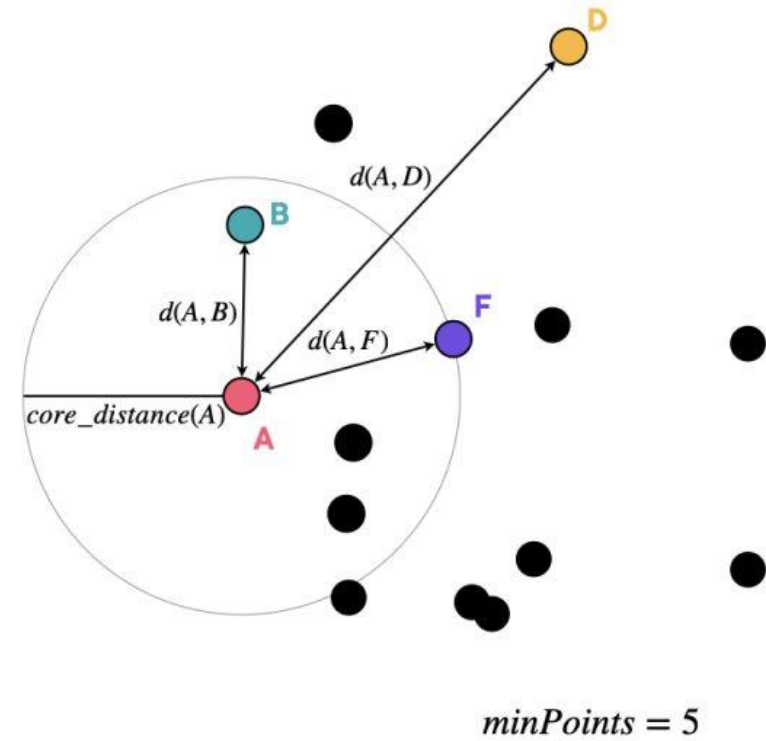


Density-Based Clustering

OPTICS (Ordering Points to Identify the Clustering Structure) : Extension of DBSCAN

- OPTICS juga berfokus pada kluster dengan mempertimbangkan struktur kerapatan data. Ini menghasilkan urutan pengamatan yang diurutkan berdasarkan tingkat kerapatan, yang kemudian dapat digunakan untuk mengidentifikasi kluster.
- Tidak memerlukan pengaturan parameter global seperti DBSCAN, dan memberikan fleksibilitas lebih besar dalam mengidentifikasi kluster pada tingkat kerapatan yang berbeda.
- Optic menggunakan jarak epsilon dan MinPoints seperti DBSCAN, dan menambahkan 2 konsep yaitu Core Distance dan Reachability Distance
- Core Distance adalah jarak minimum yang diperlukan agar suatu titik ditetapkan sebagai core point
- Reachability Distance adalah jarak terkecil dari suatu titik dan core point sehingga dapat dijangkau secara langsung dari core point/titik pusat

Core Distance dan Rechability Distance



$$\left. \begin{array}{l} d(A, F) = 0.4489609 = core_distance(A) \\ d(A, B) = 0.3452897 < core_distance(A) \\ d(A, D) = 0.9740773 > core_distance(A) \end{array} \right\} \Rightarrow \begin{array}{l} reachability_distance(B, A) = core_distance(A) = 0.4489609 \\ reachability_distance(F, A) = core_distance(A) = 0.4489609 \\ reachability_distance(D, A) = d(A, D) = 0.9740773 \end{array}$$

Titik F adalah anggota terjauh agar titik A memenuhi MinPoints. Jarak titik A dan titik F adalah Core Distance. Sedangkan jarak titik A ke semua titik adalah Rechability Distance

Cara kerja OPTICS

- OPTICS akan menghitung core distance dari titik awal
- Jika memenuhi MinPoints, OPTICS akan menghitung reachability distance anggotanya, menyimpan dan mengurutkannya dari yang terkecil
- Selanjutnya OPTICS berpindah ke titik yang memiliki reachability distance paling kecil dan menghitung core distancenya dan reachability distance setiap anggota
- Jika reachability distance lebih kecil dari sebelumnya, daftar reachability distance diperbaharui dan diurutkan kembali
- Langkah-langkah tersebut diulangi kembali sampai semua titik terjangkau
- Cluster diidentifikasi dengan mengelompokkan titik-titik yang memiliki reachability distance lebih kecil dari jarak epsilon

Multi-Step Clustering

1 SAX Transformation (Symbolic Aggregate approXimation)


- SAX adalah metode transformasi data time series menjadi simbol-simbol yang merepresentasikan bagian-bagian dari deret waktu.
- Menggunakan teknik discretization untuk mengubah nilai-nilai dalam deret waktu menjadi simbol-simbol.
- Memungkinkan analisis lebih lanjut dengan menggunakan pendekatan berbasis simbol.

2 CAST (Clustering After Sequence Transformation)

- CAST adalah metode klasterisasi yang dirancang khusus untuk deret waktu yang diubah menggunakan teknik seperti SAX.
- Menerapkan algoritma klasterisasi setelah transformasi deret waktu menjadi representasi simbolik.
- Memfasilitasi analisis klasterisasi pada data time series yang telah diubah.

3 DTW (Dynamic Time Warping)

- DTW adalah metode yang digunakan untuk mengukur kesamaan antara dua deret waktu yang mungkin berbeda dalam skala waktu.
- Menghitung "warping cost" untuk mengukur sejauh mana dua deret waktu dapat disesuaikan satu sama lain.
- Sering digunakan dalam pencarian pola dan klasterisasi deret waktu dengan ketidakpastian terhadap perbedaan skala dan waktu.



Kesimpulan dan Poin Penting

Terakhir, kita telah membahas berbagai algoritma klasterisasi time series yang dapat digunakan untuk menganalisis dan mengelompokkan data menggunakan bahasa R. Dengan pemahaman yang mendalam tentang algoritma-algoritma tersebut, Anda dapat mengambil langkah-langkah yang sesuai untuk klasterisasi data time series Anda dan mendapatkan wawasan berharga dari analisis tersebut.