Bab 08 — Unsupervised dan Representation Learning

Cara membaca bab ini

Bab 8 mengikuti pola Bab 7: setiap bagian adalah subbab, bukan unit baca kaku. Setiap subbab punya cerita, intuisi, rumus yang diperkenalkan pelan-pelan, contoh hitung, cara membaca gambar, dan tes cepat. Fokus bab ini adalah belajar dari data tanpa label: bagaimana komputer menemukan kelompok, struktur, arah variasi, representasi, dan anomali ketika kita belum memiliki jawaban benar.

Bab ini penting karena dunia nyata sering tidak menyediakan label rapi. Toko punya ribuan transaksi tetapi belum tahu segmen pelanggan. Sekolah punya catatan aktivitas belajar tetapi belum tahu pola kesulitan siswa. Aplikasi punya log penggunaan tetapi belum tahu perilaku mana yang normal atau mencurigakan. Di sinilah unsupervised learning membantu: bukan memberi jawaban final, tetapi membuka peta.

Subbab 1 — Apa itu unsupervised learning?

Inti subbab: unsupervised learning mencari struktur dalam data tanpa label target y.

Pada supervised learning, data berbentuk pasangan (x, y): fitur dan jawaban. Pada unsupervised learning, kita hanya punya fitur:

$$X = \{x_1, x_2, \ldots, x_n\}$$

Tidak ada kolom “benar” yang langsung memberi tahu model. Tugas model adalah menemukan pola: data mana yang mirip, arah variasi mana yang paling penting, titik mana yang aneh, atau representasi ringkas apa yang menyimpan informasi utama.

Contoh Indonesia: sebuah warung kopi punya data pelanggan:

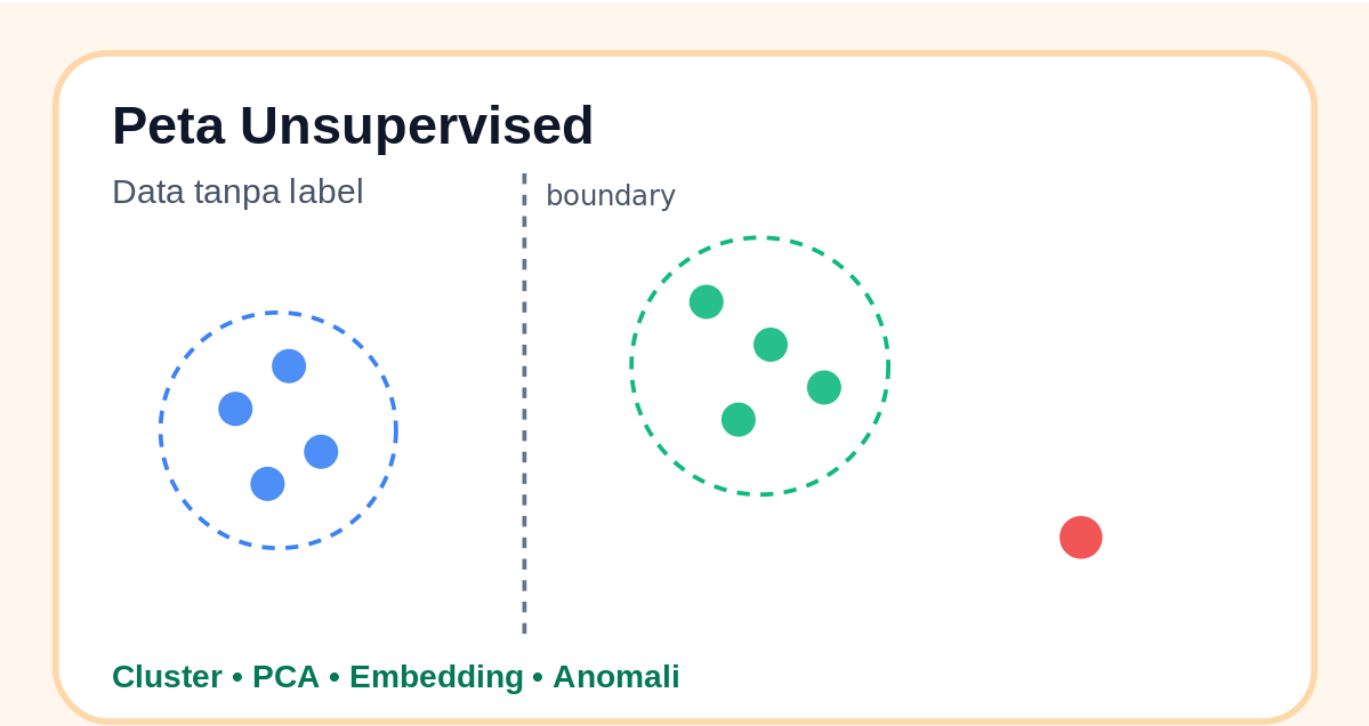
```
x = [jumlah_kunjungan_per_bulan, rata_rata_belanja, persen_pesan_kopi_susu]
```

Pemilik belum punya label “pelanggan hemat”, “pelanggan loyal”, atau “pelanggan promo”. Unsupervised learning bisa membantu mengelompokkan pelanggan berdasarkan pola perilaku. Namun hasil cluster bukan kebenaran otomatis. Cluster harus ditafsirkan manusia: apakah kelompok itu masuk akal, adil, dan berguna?

Secara umum:

```
input:  X tanpa label y
output: struktur z, cluster c, embedding h, atau skor anomali s
```

Cara membaca gambar: perhatikan titik data sebagai pelanggan. Warna bukan label asli, tetapi hasil dugaan kelompok. Jika warna terlihat terpisah, struktur mungkin kuat. Jika tumpang tindih, model perlu diperiksa.



Peta unsupervised learning

Contoh hitung mini

Tiga pelanggan direpresentasikan oleh dua fitur:

A = [10 kunjungan, 50 ribu]
B = [11 kunjungan, 52 ribu]
C = [2 kunjungan, 15 ribu]

Secara intuisi A dan B lebih mirip daripada A dan C. Bab ini akan membuat intuisi itu menjadi jarak, cluster, dan representasi.

Tes cepat subbab 1

1. Apa perbedaan utama supervised dan unsupervised learning?

2. Mengapa hasil cluster tidak boleh langsung dianggap “kebenaran”?

3. Sebutkan satu contoh lokal data tanpa label.

Subbab 2 — Sejarah singkat: dari taksonomi sampai representasi modern

Inti subbab: unsupervised learning berkembang dari statistik, psikometri, taksonomi, kompresi data, sampai deep representation learning.

Sebelum komputer modern, manusia sudah melakukan pengelompokan: ahli biologi membuat taksonomi spesies, perpustakaan mengelompokkan buku, pedagang mengelompokkan pelanggan, dan ilmuwan sosial mencari pola dari survei. Dalam statistik, metode seperti principal component analysis (PCA) muncul awal abad ke-20 untuk merangkum banyak variabel menjadi beberapa arah utama. Dalam machine learning modern, clustering, manifold learning, embeddings, autoencoder, dan self-supervised learning memperluas ide yang sama: temukan struktur yang tidak tertulis eksplisit.

Timeline ringkas:

1901: Pearson memperkenalkan gagasan principal components

1930-an: Hotelling mengembangkan PCA lebih formal
1950-an–1960-an: k-means dan hierarchical clustering populer dalam statistik komputasional
1990-an: kernel methods, spectral methods, manifold learning berkembang
2010-an: word embeddings dan deep representation learning naik
2020-an: self-supervised learning menjadi fondasi banyak model besar

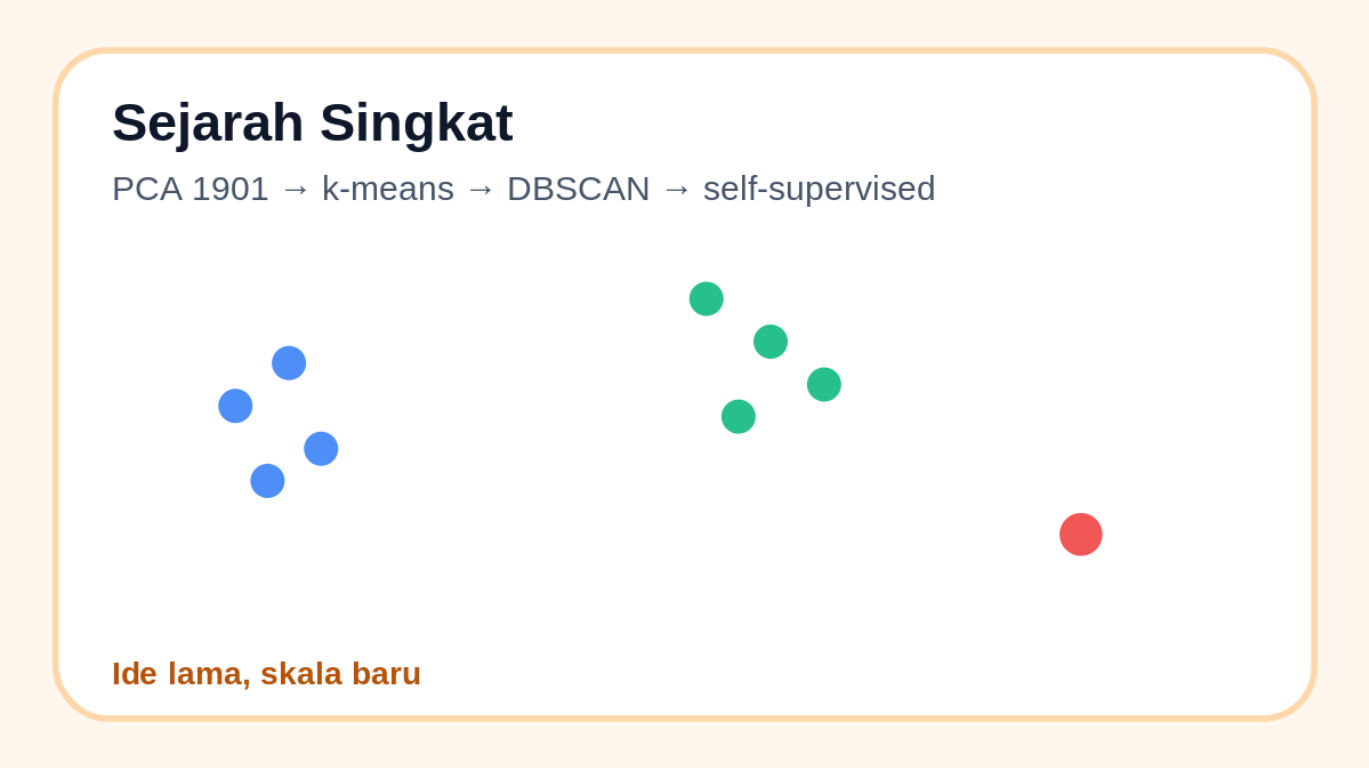
Pelajaran sejarahnya: unsupervised learning bukan satu algoritma, melainkan keluarga cara berpikir. K-means mencari pusat kelompok. PCA mencari arah variasi. Embedding mencari ruang representasi. Anomaly detection mencari titik yang tidak biasa. Semua berbagi pertanyaan: “struktur apa yang tersembunyi di dalam data?”

Persamaan payung

Banyak metode unsupervised dapat dibaca sebagai optimisasi:

```
struktur* = argmin_struktur objective(X, struktur)
```

K-means meminimalkan jarak titik ke centroid. PCA memaksimalkan variansi yang dijelaskan. Autoencoder meminimalkan error rekonstruksi.



Sejarah unsupervised learning

Contoh interpretasi

Jika sebuah metode memberi dua cluster pelanggan, jangan berhenti pada “cluster 0” dan “cluster 1”. Beri nama berdasarkan ciri: “sering datang, belanja sedang” atau “jarang datang, belanja tinggi”. Nama ini bagian dari analisis, bukan keluaran otomatis model.

Tes cepat subbab 2

1. Mengapa PCA bisa dianggap metode unsupervised?

2. Apa kesamaan k-means, PCA, dan autoencoder dari sudut optimisasi?

3. Mengapa sejarah unsupervised learning lebih tua daripada LLM modern?

Subbab 3 — Dataset tanpa label dan notasi matriks X

Inti subbab: data unsupervised biasanya ditulis sebagai matriks `X` berukuran `n × d`.

Jika ada n baris data dan d fitur, kita tulis:

$X \in R^{(n \times d)}$

Baris ke-i adalah contoh data:

$x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$

Contoh dataset warung:

pelanggan	kunjungan	belanja	persen kopi susu
A	10	50	80
B	11	52	75
C	2	15	10
D	3	17	20

Matriks fiturnya:

```
X = [[10, 50, 80],
     [11, 52, 75],
     [ 2, 15, 10],
     [ 3, 17, 20]]
```

Tidak ada kolom label seperti “loyal” atau “tidak loyal”. Jika nanti k-means memberi cluster `[0,0,1,1]`, angka itu bukan label asli; itu hasil konstruksi model.

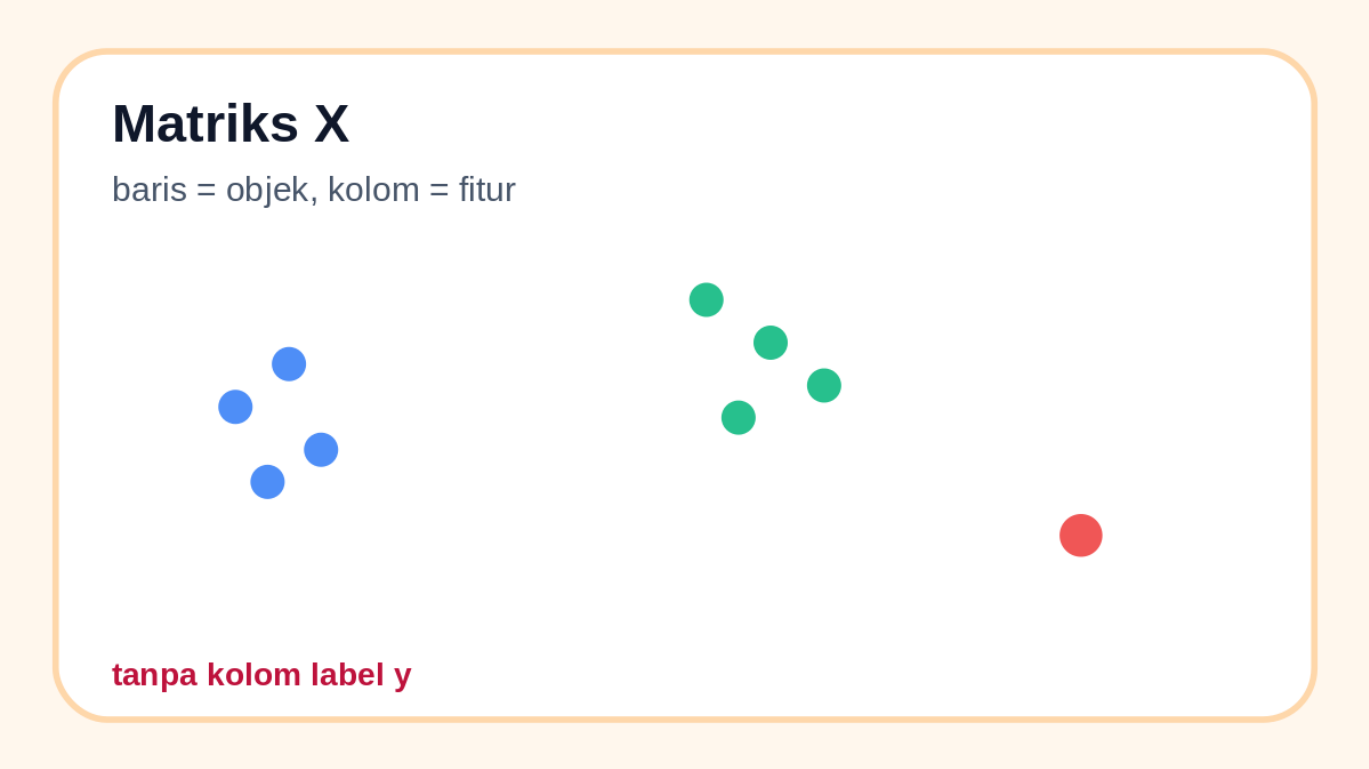
Contoh hitung: dimensi dan baris

Dataset di atas punya 4 pelanggan dan 3 fitur, jadi:

$n = 4, d = 3, X \in R^{(4 \times 3)}$

Baris pelanggan B:

$x_B = [11, 52, 75]$



Matriks data tanpa label

Cara membaca gambar: baris adalah objek, kolom adalah fitur. Dalam unsupervised learning, tidak ada kolom jawaban. Model bekerja hanya dari hubungan antarbaris dan antarkolom.

Tes cepat subbab 3

1. Jika dataset punya 200 pelanggan dan 5 fitur, berapa ukuran matriks X?

2. Mengapa angka cluster hasil model tidak sama dengan label asli?

3. Apa risiko jika fitur penting tidak dimasukkan ke matriks?

Subbab 4 — Data preparing: dari data mentah ke data siap dianalisis

Inti subbab: sebelum clustering, PCA, atau anomaly detection, data harus disiapkan. Model yang bagus tidak bisa menyelamatkan data yang kacau.

Di proyek AI nyata, bagian paling lama sering bukan training model, melainkan memahami data. Data mentah biasanya punya nilai kosong, satuan berbeda, duplikasi, typo, kategori tidak konsisten, outlier, dan kolom yang tampak angka tetapi sebenarnya kode. Karena itu, pipeline Bab 8 diperluas:

data mentah → audit skema → cleansing → preprocessing → visualisasi → insight → model

Cara membaca pipeline: panah → dibaca “lanjut ke”. Artinya data mentah tidak langsung masuk model. Ia melewati pemeriksaan skema, pembersihan, transformasi, visualisasi, baru kemudian dipakai untuk model.

Contoh data pelanggan mentah:

nama	kunjungan	belanja	kopi_susu_%	catatan
Ayu	10	50	80	ok
Bima	11	52	75	ok
Citra	2	15	10	ok
Dedi	-3	17	20	kunjungan salah
Eka	9	kosong	78	missing
Ayu	10	50	80	duplikat

Sebelum model, kita perlu bertanya:

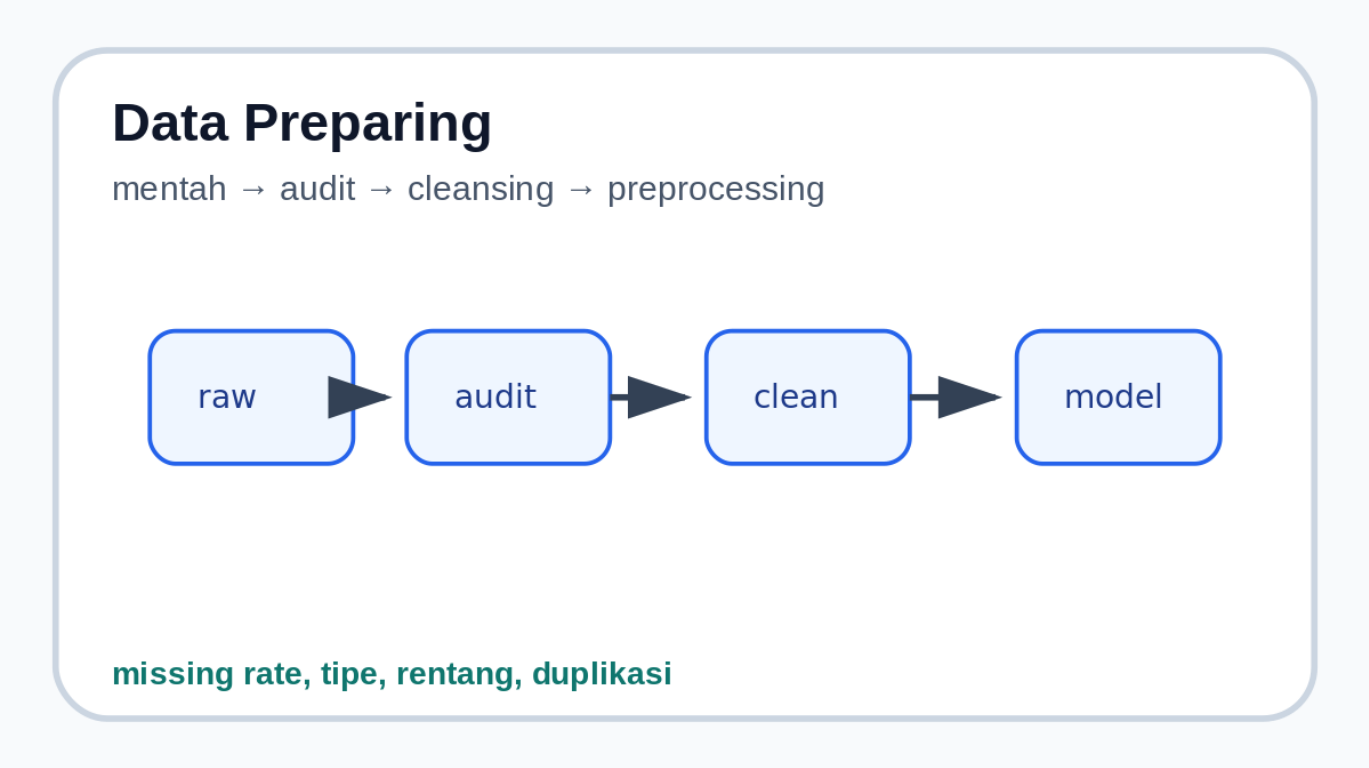
Apakah tipe data benar?
Apakah rentang nilai masuk akal?
Apakah ada nilai kosong?
Apakah ada duplikasi?
Apakah satuan fitur konsisten?
Apakah ada kolom bocor atau tidak relevan?

Contoh hitung audit missing value

Jika ada 6 baris dan 1 nilai belanja kosong:

missing_rate = jumlah_nilai_kosong / jumlah_baris
= 1 / 6
= 0,1667 ≈ 16,67%

Cara membaca rumus: `missing_rate` dibaca “tingkat kekosongan”. Pembilang adalah jumlah nilai kosong. Penyebut adalah jumlah baris. Jika hasilnya besar, kualitas data perlu diperbaiki sebelum model.



Data preparing pipeline

Keputusan praktis

• Nilai kosong sedikit: bisa imputasi sederhana, misalnya median.

• Nilai negatif pada fitur yang mustahil negatif: perlu koreksi atau buang baris.

• Duplikasi identik: biasanya hapus salah satu.

• Satuan campur, misalnya ribu dan rupiah penuh: samakan satuan.

Tes cepat subbab 4

1. Mengapa data mentah tidak boleh langsung dimasukkan ke k-means?

2. Hitung missing rate jika 8 dari 100 baris punya nilai kosong.

3. Sebutkan dua contoh nilai yang harus dicurigai pada data pelanggan.

Subbab 5 — Data cleansing: missing value, duplikasi, outlier, dan validasi rentang

Inti subbab: data cleansing mengubah data yang kacau menjadi data yang cukup jujur untuk dianalisis.

Cleansing bukan manipulasi agar hasil terlihat bagus. Cleansing adalah proses membuat data lebih sesuai dengan realitas. Kita harus mencatat apa yang diubah agar eksperimen bisa diaudit.

Empat operasi dasar:

1. Validasi tipe: angka tetap angka, tanggal tetap tanggal.
2. Validasi rentang: kunjungan tidak boleh negatif.
3. Duplikasi: baris identik jangan dihitung dua kali tanpa alasan.
4. Missing value: isi, tandai, atau buang dengan alasan jelas.

Imputasi median

Jika data belanja:

```
[15, 17, kosong, 50, 52, 90]
```

Median dari nilai yang ada:

```
urut = [15, 17, 50, 52, 90]
median = 50
```

Nilai kosong bisa diisi 50 jika konteksnya masuk akal.

Cara membaca rumus median: median adalah nilai tengah setelah data diurutkan. Jika jumlah data ganjil, ambil titik tengah. Jika genap, biasanya rata-rata dua nilai tengah.

Outlier dengan IQR

Metode IQR memakai kuartil:

```
IQR = Q3 - Q1
batas_bawah = Q1 - 1,5×IQR
batas_atas  = Q3 + 1,5×IQR
```

Cara membaca rumus: `Q1` adalah kuartil bawah, `Q3` kuartil atas. IQR mengukur rentang tengah data. Nilai di luar batas belum pasti salah, tetapi perlu diperiksa.

Contoh:

```
Q1=20, Q3=60
IQR = 60-20 = 40
batas_atas = 60 + 1,5×40 = 120
```

Nilai belanja 170 berada di atas 120, maka masuk kandidat outlier.

Data cleansing

Catatan etika: jangan menghapus outlier hanya karena mengganggu model. Outlier bisa fraud, pelanggan VIP, sensor rusak, atau kejadian penting.

Tes cepat subbab 5

1. Apa beda missing value dan outlier?

2. Hitung IQR jika `Q1=10` dan `Q3=22`.

3. Mengapa cleansing harus dicatat dalam laporan eksperimen?

Subbab 6 — Data visualization dan EDA: melihat pola sebelum model

Inti subbab: visualisasi data membantu menemukan pola, kesalahan, anomali, dan pertanyaan baru sebelum model dilatih.

EDA (*exploratory data analysis*) adalah proses eksplorasi. Di Bab 3 pembaca sudah belajar Python dan data mini. Di Bab 8, EDA dipakai untuk kasus unsupervised: mencari struktur tanpa label. Visualisasi yang paling berguna untuk awal:

```
histogram     → distribusi satu fitur
scatter plot  → hubungan dua fitur
boxplot       → median, kuartil, kandidat outlier
line plot     → perubahan terhadap waktu
heatmap       → korelasi atau kemiripan antarfitur
```

Histogram menjawab: nilai paling sering ada di mana? Apakah distribusi miring? Ada dua puncak? Ada ekor panjang?

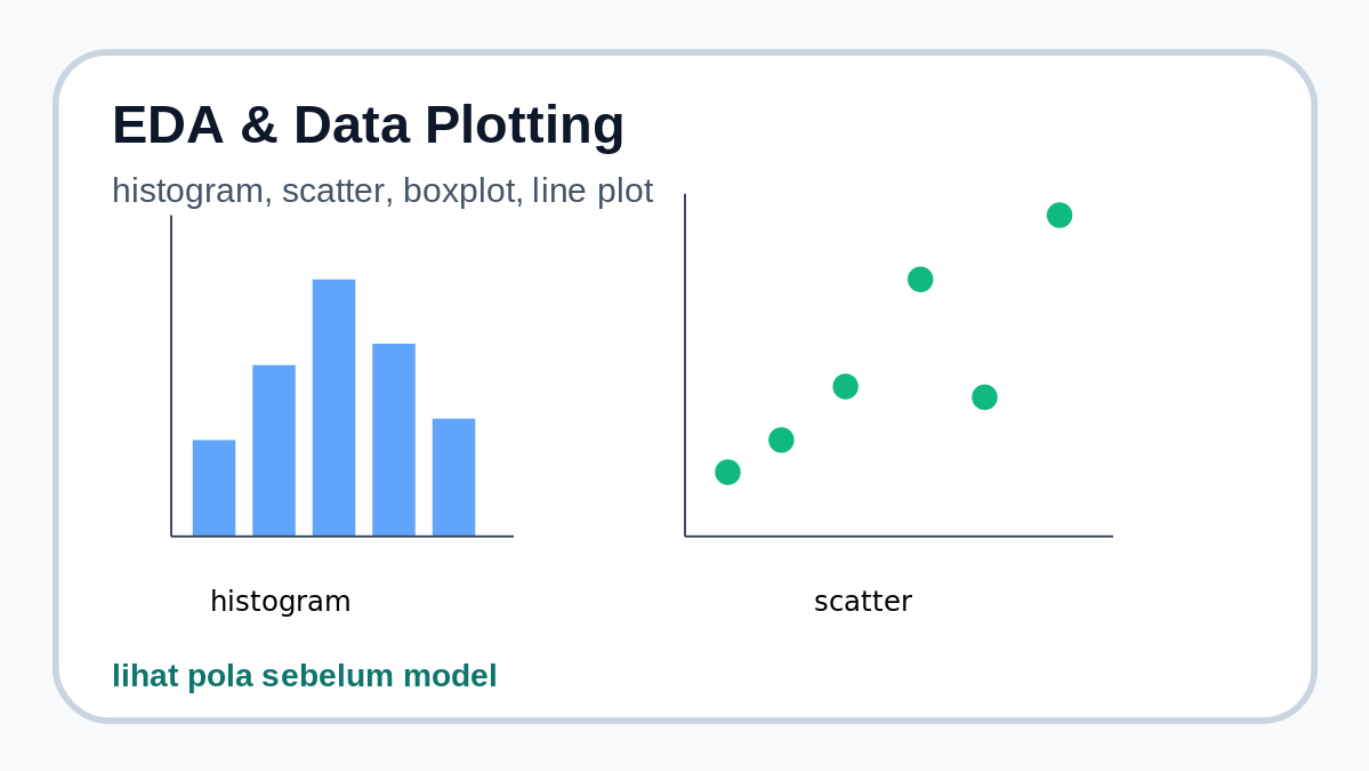
Scatter plot menjawab: apakah dua fitur naik bersama, turun bersama, membentuk cluster, atau punya anomali?

Contoh insight dari scatter

Jika `kunjungan` dan `belanja` diplot, pola yang mungkin muncul:

1. Titik membentuk garis naik → makin sering datang, makin besar belanja.
2. Titik membentuk dua awan → mungkin ada dua segmen pelanggan.
3. Satu titik jauh sendiri → kandidat anomali atau pelanggan khusus.

Cara membaca grafik scatter: sumbu-X adalah fitur pertama, sumbu-Y fitur kedua. Setiap titik adalah satu objek. Jarak titik membantu membaca kemiripan, tetapi hanya untuk dua fitur yang diplot.



Data plotting dan EDA

Contoh hitung ringkasan EDA

Data belanja `[15, 17, 48, 50, 52, 90]`:

mean = (15+17+48+50+52+90)/6 = 272/6 = 45,33
min = 15
max = 90
range = 90-15 = 75

Cara membaca rumus mean: jumlahkan semua nilai, lalu bagi dengan banyak data. Mean sensitif terhadap nilai ekstrem, sehingga perlu dibandingkan dengan median.

Tes cepat subbab 6

1. Kapan scatter plot lebih berguna daripada histogram?

2. Apa arti titik yang jauh sendiri pada scatter plot?

3. Hitung range dari data `[4, 8, 10, 20]`.

Subbab 7 — Regresi linear sebagai alat insight sebelum clustering

Inti subbab: walaupun regresi linear termasuk supervised learning, garis regresi sederhana berguna untuk EDA: melihat tren, residual, dan anomali.

Pembahasan utama regresi ada di Bab 7, tetapi Bab 8 membutuhkan satu alat penting: garis tren. Saat melihat scatter plot `kunjungan` vs `belanja`, kita bisa bertanya: apakah belanja cenderung naik ketika kunjungan naik?

Regresi linear satu fitur:

$$\hat{y} = wx + b$$

Cara membaca rumus: ■ dibaca “y topi” atau prediksi y. `x` adalah fitur input. `w` adalah kemiringan garis. `b` adalah intercept, yaitu nilai prediksi saat `x=0`.

Rumus slope closed-form:

$$w = \Sigma(x_i - \bar{x})(y_i - \bar{y}) / \Sigma(x_i - \bar{x})^2$$
$$b = \bar{y} - w\,\bar{x}$$

Cara membaca rumus: x■ dibaca “x bar”, rata-rata x. ■ dibaca “y bar”, rata-rata y. Pembilang mengukur apakah x dan y bergerak bersama. Penyebut mengukur variasi x. Jika `w` positif, y cenderung naik saat x naik.

Contoh hitung kecil

Data:

x = [1, 2, 3]
y = [2, 4, 6]

Rata-rata:

$\bar{x} = (1+2+3)/3 = 2$
$\bar{y} = (2+4+6)/3 = 4$

Pembilang:

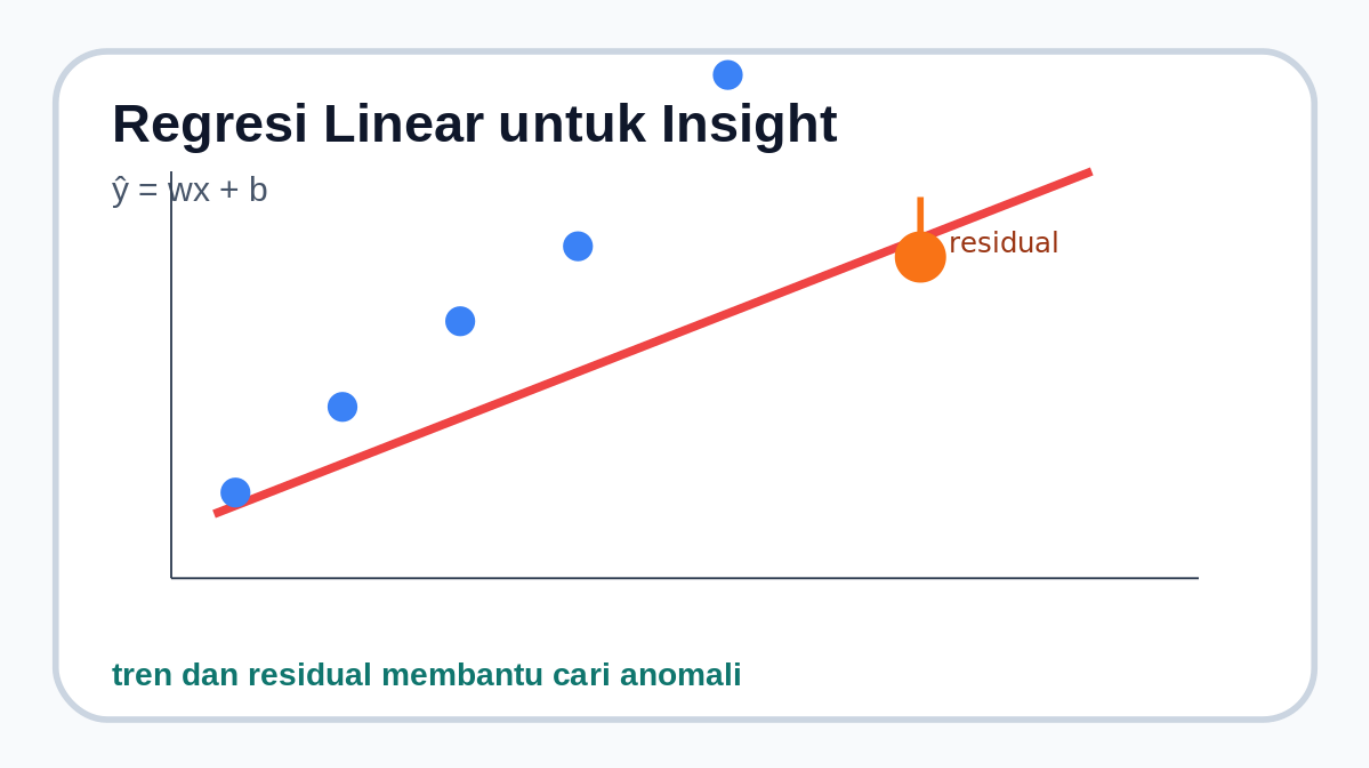
(1-2)(2-4) + (2-2)(4-4) + (3-2)(6-4)
= (-1)(-2) + 0·0 + 1·2
= 2 + 0 + 2 = 4

Penyebut:

$(1-2)^2 + (2-2)^2 + (3-2)^2$
$= 1 + 0 + 1 = 2$

Maka:

w = 4/2 = 2
b = 4 - 2×2 = 0
$\hat{y} = 2x$



Regresi linear untuk insight

Residual untuk anomali

Residual:

$$e_i = y_i - \hat{y}_i$$

Cara membaca rumus: residual adalah selisih antara nilai aktual dan prediksi garis. Residual besar berarti titik jauh dari tren umum. Dalam EDA, residual besar bisa menjadi kandidat anomali.

Contoh: jika garis memprediksi belanja 40, tetapi aktual 90:

$$e = 90 - 40 = 50$$

Titik ini perlu diperiksa: mungkin pelanggan khusus, data salah, atau event tertentu.

Tes cepat subbab 7

1. Apa arti w pada regresi linear?

2. Hitung prediksi ■ jika w=2, b=3, x=5.

3. Mengapa residual besar berguna untuk mencari anomali?

Subbab 8 — Jarak dan kemiripan: bahasa dasar pengelompokan

Inti subbab: banyak metode unsupervised bergantung pada definisi “mirip”.

Jika dua titik dekat, kita sering menganggap keduanya mirip. Jarak Euclidean dua titik a dan b:

$$d(a,b) = \text{sqrt}(\Sigma_j (a_j - b_j)^2)$$

Jarak Manhattan:

$$d_1(a,b) = \Sigma_j |a_j - b_j|$$

Cosine similarity:

$$\cos(a,b) = (a \cdot b) / (||a|| \, ||b||)$$

Euclidean cocok untuk ruang geometris. Manhattan sering intuitif untuk grid/kota. Cosine sering dipakai untuk teks dan embedding karena lebih peduli arah daripada panjang.

Contoh hitung Euclidean

A = [10, 50]
B = [11, 52]
C = [2, 15]

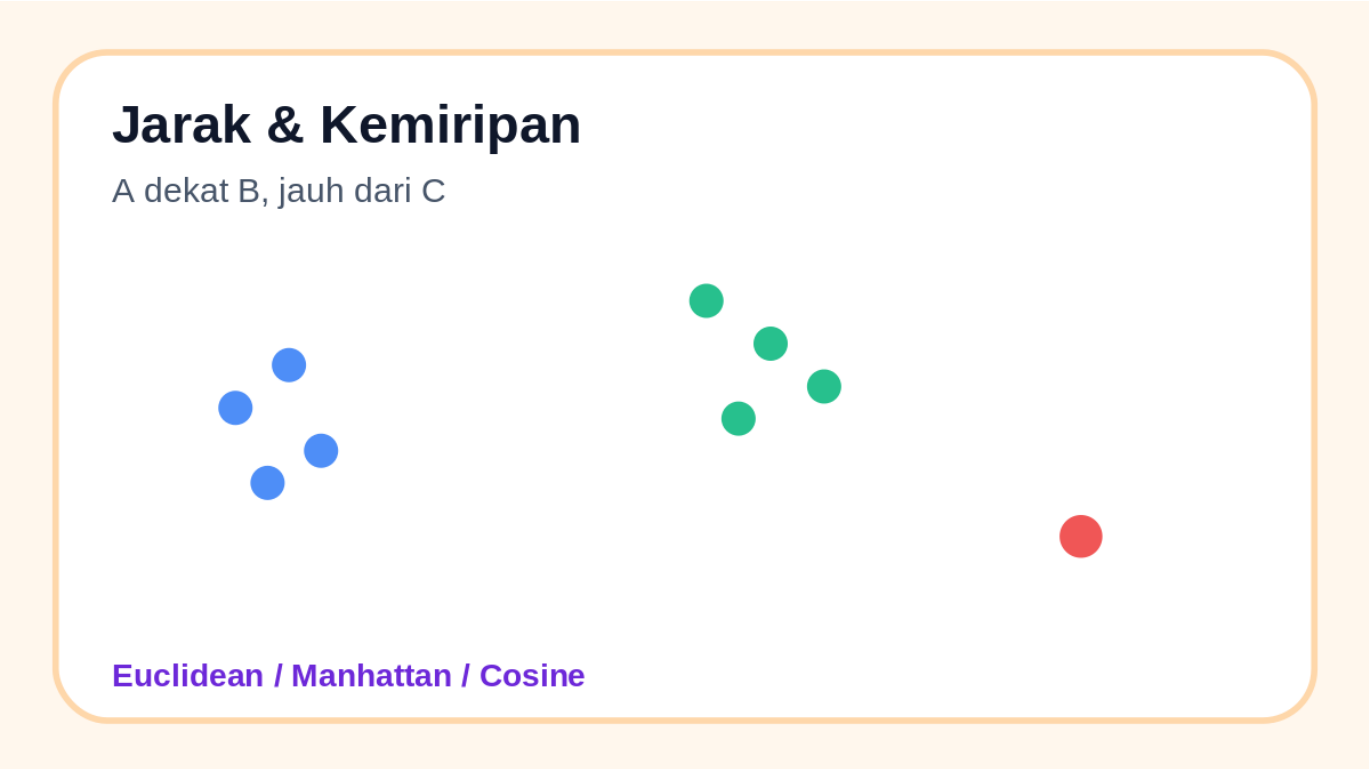
Jarak A-B:

sqrt((10-11)² + (50-52)²)
= sqrt(1 + 4)
= sqrt(5)
≈ 2,24

Jarak A-C:

sqrt((10-2)² + (50-15)²)
= sqrt(64 + 1225)
= sqrt(1289)
≈ 35,90

A lebih dekat ke B daripada ke C.



Jarak antar titik

Catatan penting: definisi jarak adalah keputusan desain. Dua pelanggan bisa dekat dari sisi belanja, tetapi jauh dari sisi preferensi produk. Model hanya melihat fitur yang kita berikan.

Tes cepat subbab 8

1. Hitung jarak Euclidean antara `[3,4]` dan `[0,0]`.

2. Mengapa cosine similarity populer untuk embedding teks?

3. Apa yang terjadi jika kita memilih metrik jarak yang tidak cocok?

Subbab 9 — Scaling: agar fitur besar tidak menindas fitur kecil

Inti subbab: sebelum clustering/PCA, fitur perlu diskalakan agar ukuran angka tidak mendominasi makna.

Misalkan fitur `belanja_rupiah` bernilai puluhan ribu, sedangkan `jumlah_kunjungan` bernilai 1 sampai 20. Jika langsung memakai Euclidean, fitur rupiah bisa mendominasi jarak. Salah satu solusi adalah standardisasi:

$$z = (x - \mu) / \sigma$$

μ adalah rata-rata fitur, σ adalah standar deviasi. Setelah standardisasi, fitur dibaca sebagai "berapa standar deviasi dari rata-rata".

Contoh hitung sederhana

Data kunjungan: `[10, 12, 14]`

$$\mu = (10+12+14)/3 = 12$$

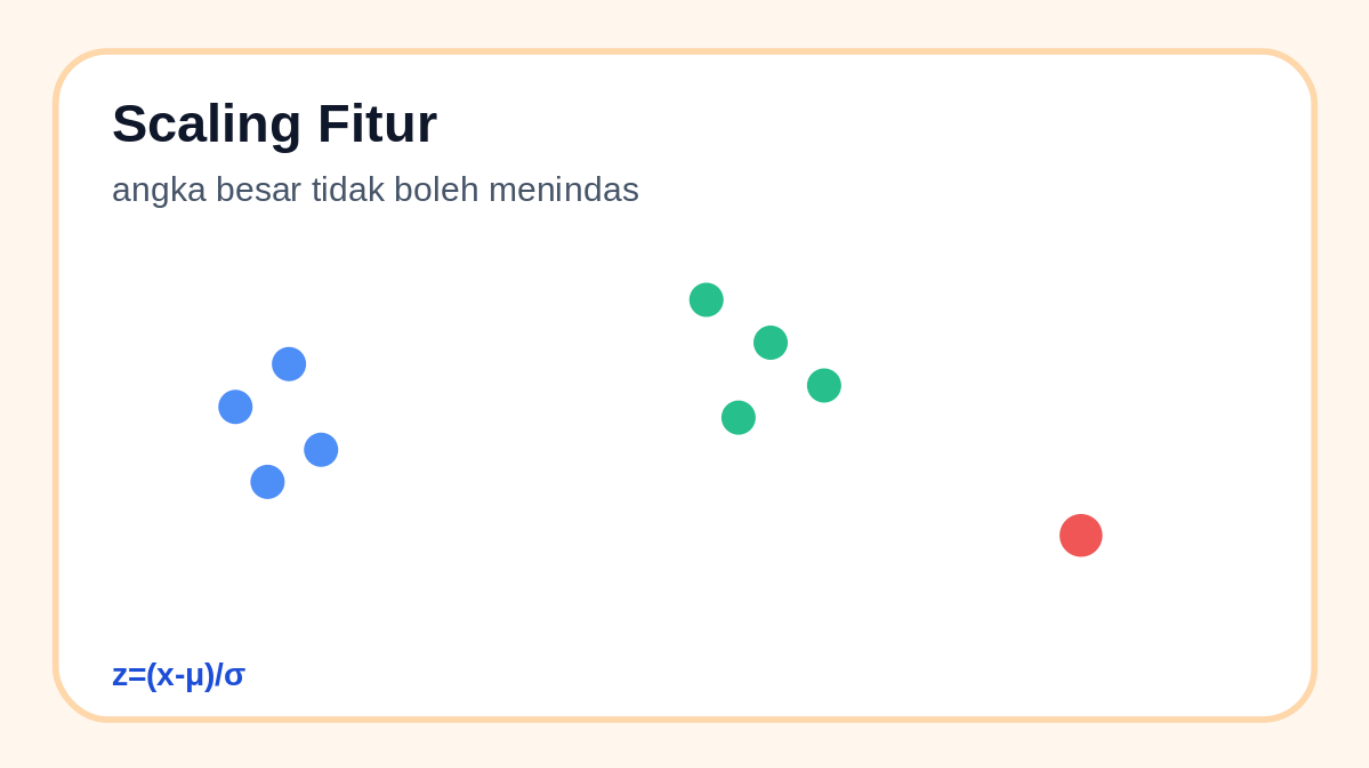
Untuk memudahkan, anggap standar deviasi populasi:

$$\sigma = \text{sqrt}(((10-12)^2 + (12-12)^2 + (14-12)^2)/3)$$
$$= \text{sqrt}((4+0+4)/3)$$
$$= \text{sqrt}(2,667)$$
$$\approx 1,633$$

Z-score untuk 14:

$$z = (14-12)/1,633 \approx 1,225$$

Artinya nilai 14 berada sekitar 1,225 standar deviasi di atas rata-rata.



Scaling fitur

Kesalahan umum: melakukan scaling memakai seluruh data termasuk data evaluasi. Untuk eksperimen serius, parameter scaling dihitung dari data training lalu diterapkan ke data lain.

Tes cepat subbab 9

1. Mengapa fitur rupiah bisa mendominasi jarak?
2. Hitung z-score untuk $x=10$, $\mu=12$, $\sigma=2$.
3. Kapan scaling tidak selalu diperlukan?

Subbab 10 — Clustering: memberi struktur awal pada data

Inti subbab: clustering mengelompokkan titik yang mirip, tetapi nama dan makna cluster harus dianalisis manusia.

Clustering mencoba membuat partisi:

$$C = \{C_1, C_2, ..., C_k\}$$

Setiap C■ adalah kelompok data. Tujuannya bukan sekadar mewarnai titik, tetapi menemukan struktur yang membantu keputusan. Dalam bisnis, cluster pelanggan bisa membantu strategi layanan. Dalam pendidikan, cluster aktivitas belajar bisa membantu intervensi. Dalam keamanan, cluster log bisa mengungkap pola perilaku.

Namun clustering rentan disalahpahami. Jika model menemukan 3 cluster, bukan berarti dunia benar-benar punya 3 kelompok alami. Jumlah cluster bisa dipengaruhi scaling, fitur, algoritma, dan noise.

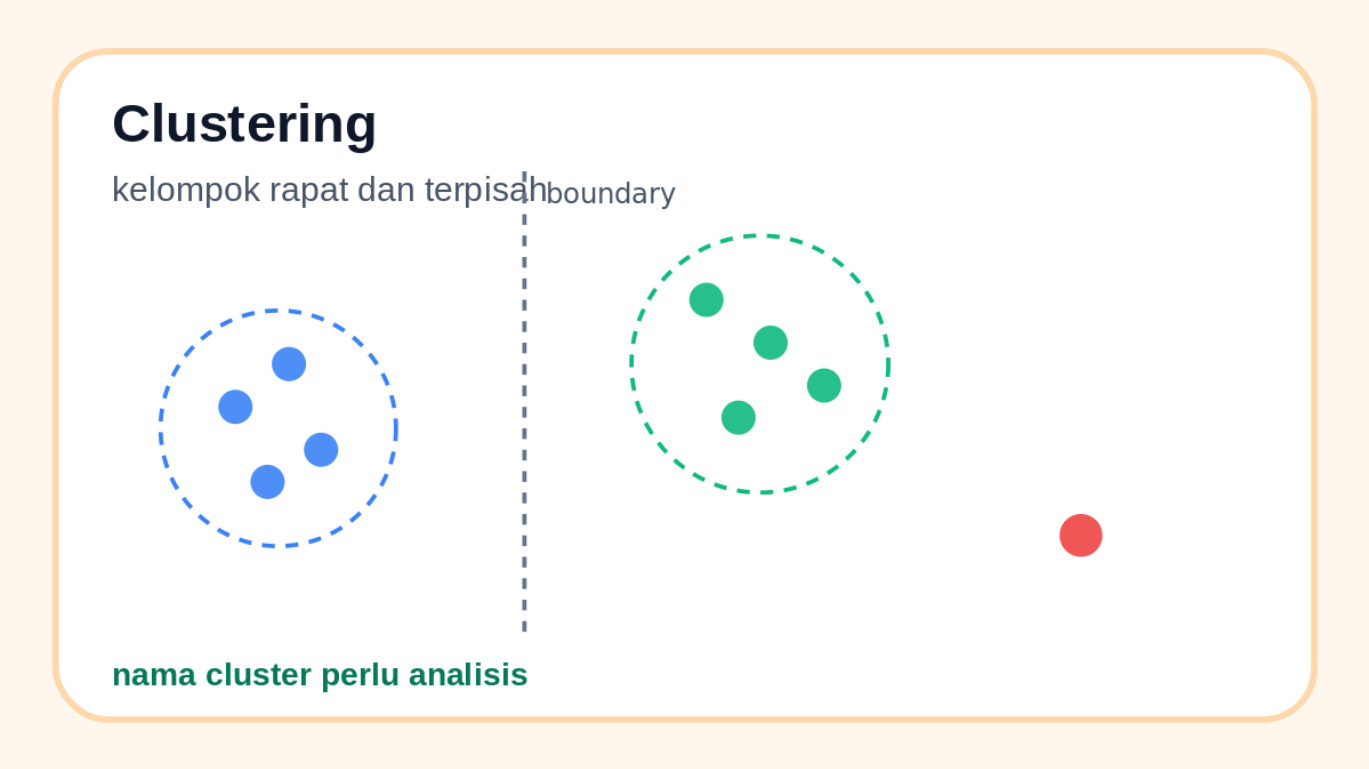
Contoh interpretasi cluster

Misalkan k-means menghasilkan:

Cluster 0: kunjungan tinggi, belanja sedang
Cluster 1: kunjungan rendah, belanja rendah
Cluster 2: kunjungan rendah, belanja tinggi

Nama cluster yang lebih berguna:

Cluster 0 → pelanggan rutin
Cluster 1 → pelanggan pasif
Cluster 2 → pembeli sesekali bernilai tinggi



Intuisi clustering

Cara membaca gambar: lihat jarak antar titik dalam kelompok dan jarak antarkelompok. Cluster yang baik biasanya rapat di dalam dan terpisah di luar, tetapi data dunia nyata sering tidak serapi ilustrasi.

Tes cepat subbab 10

1. Mengapa cluster perlu diberi nama setelah dianalisis?

2. Apa bahaya langsung menganggap cluster sebagai segmentasi final?

3. Sebutkan contoh clustering di sekolah atau UMKM.

Subbab 11 — K-means: centroid, objective, dan langkah manual

Inti subbab: k-means mencari k pusat cluster sehingga jarak titik ke pusatnya kecil.

K-means punya objective:

$$J = \Sigma_i ||x_i - \mu_{c_i}||^2$$

μ_{c■} adalah centroid cluster untuk titik x■. Algoritmanya:

1. Pilih k centroid awal
2. Assign setiap titik ke centroid terdekat
3. Update centroid = rata-rata titik dalam cluster
4. Ulangi sampai stabil

Contoh hitung satu iterasi

Data 1D:

x = [2, 4, 10, 12]
k = 2
centroid awal μ_1=2, μ_2=12

Assign:

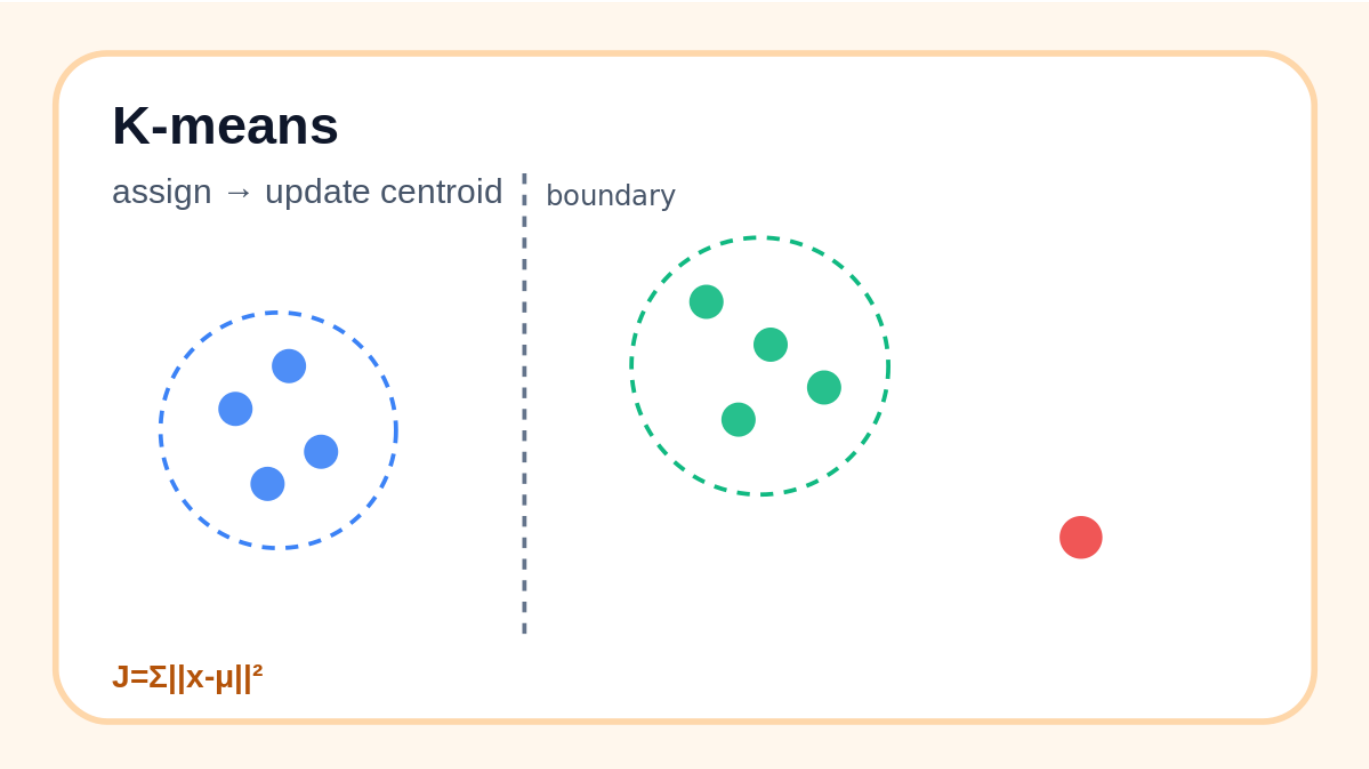
x=2: jarak ke 2 =0, ke 12=10 → cluster 1
x=4: jarak ke 2 =2, ke 12=8 → cluster 1
x=10: jarak ke 2=8, ke 12=2 → cluster 2
x=12: jarak ke 2=10, ke 12=0 → cluster 2

Update centroid:

μ_1 baru = (2+4)/2 = 3
μ_2 baru = (10+12)/2 = 11

Objective setelah update:

$J = (2-3)^2 + (4-3)^2 + (10-11)^2 + (12-11)^2$
$= 1 + 1 + 1 + 1$
$= 4$



K-means centroid

Turunan intuisi centroid

Untuk satu cluster, kita ingin meminimalkan:

$$J(\mu) = \Sigma_i (x_i - \mu)^2$$

Turunannya:

$$dJ/d\mu = \Sigma_i 2(\mu - x_i) = 2(n\mu - \Sigma_i x_i)$$

Set $dJ/d\mu = 0$:

$$n\mu = \Sigma_i x_i$$
$$\mu = (\Sigma_i x_i)/n$$

Jadi centroid optimal untuk cluster adalah rata-rata.

Tes cepat subbab 11

1. Mengapa centroid k-means adalah rata-rata?

2. Lakukan satu iterasi untuk data `[1,2,8,9]` dengan centroid awal 1 dan 9.

3. Mengapa k-means sensitif terhadap centroid awal?

Subbab 12 — Anatomi grafik k-means: titik, centroid, dan batas Voronoi

Inti subbab: diagram k-means harus dibaca sebagai peta jarak ke centroid.

Dalam grafik k-means, komponen pentingnya:

node/titik data → objek yang dikelompokkan
centroid → pusat rata-rata cluster
edge imajiner → jarak dari titik ke centroid
boundary → garis tempat dua centroid sama dekat

Batas antara dua centroid sering disebut batas Voronoi. Titik di satu sisi lebih dekat ke centroid A; titik di sisi lain lebih dekat ke centroid B.

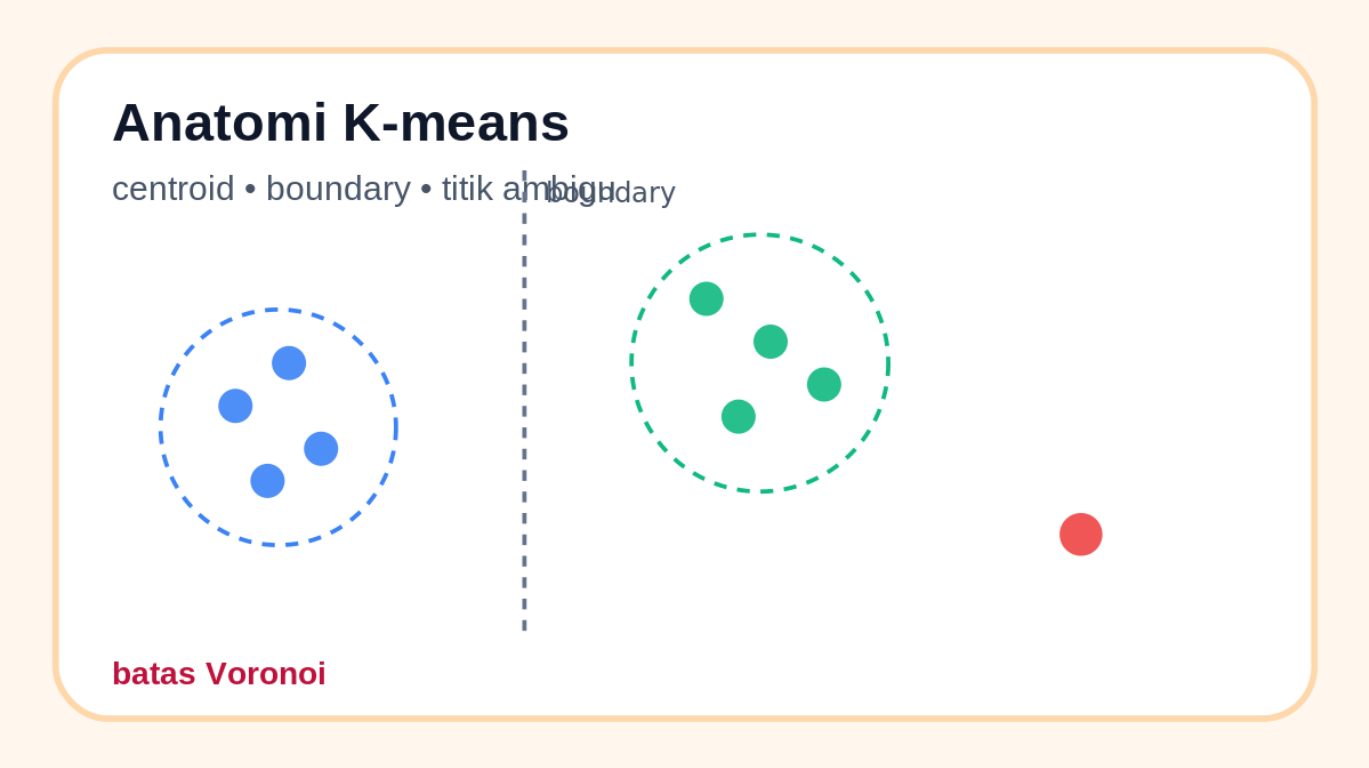
Contoh hitung batas 1D

Jika centroid berada di μ■=3 dan μ■=11, batasnya ada di titik tengah:

```
boundary = (3+11)/2 = 7
```

Titik x < 7 masuk cluster 1. Titik x > 7 masuk cluster 2.

Cara membaca gambar: jangan hanya lihat warna. Lihat pusat cluster, jarak titik ke pusat, titik yang dekat boundary, dan titik yang jauh dari semua pusat. Titik dekat boundary biasanya lebih tidak pasti.



Anatomi k-means

Latihan visual

Jika centroid A di `(0,0)` dan centroid B di `(4,0)`, titik `(2,0)` berada tepat di boundary. Titik `(1,1)` lebih dekat ke A:

```
jarak ke A = sqrt(1²+1²)=1,41
jarak ke B = sqrt((1-4)²+1²)=sqrt(10)=3,16
```

Tes cepat subbab 12

1. Apa arti centroid pada grafik k-means?

2. Mengapa titik dekat boundary perlu dianalisis hati-hati?

3. Hitung boundary 1D untuk centroid 5 dan 15.

Subbab 13 — Hierarchical clustering dan dendrogram

Inti subbab: hierarchical clustering membangun pohon penggabungan cluster.

Berbeda dari k-means yang meminta k di awal, hierarchical clustering membuat struktur bertingkat. Versi agglomerative dimulai dari setiap titik sebagai cluster sendiri, lalu

menggabungkan cluster terdekat.

Beberapa linkage:

single linkage = jarak minimum antaranggota
complete linkage = jarak maksimum antaranggota
average linkage = rata-rata jarak antaranggota

Contoh hitung kecil

Data 1D:

A=1, B=2, C=8, D=10

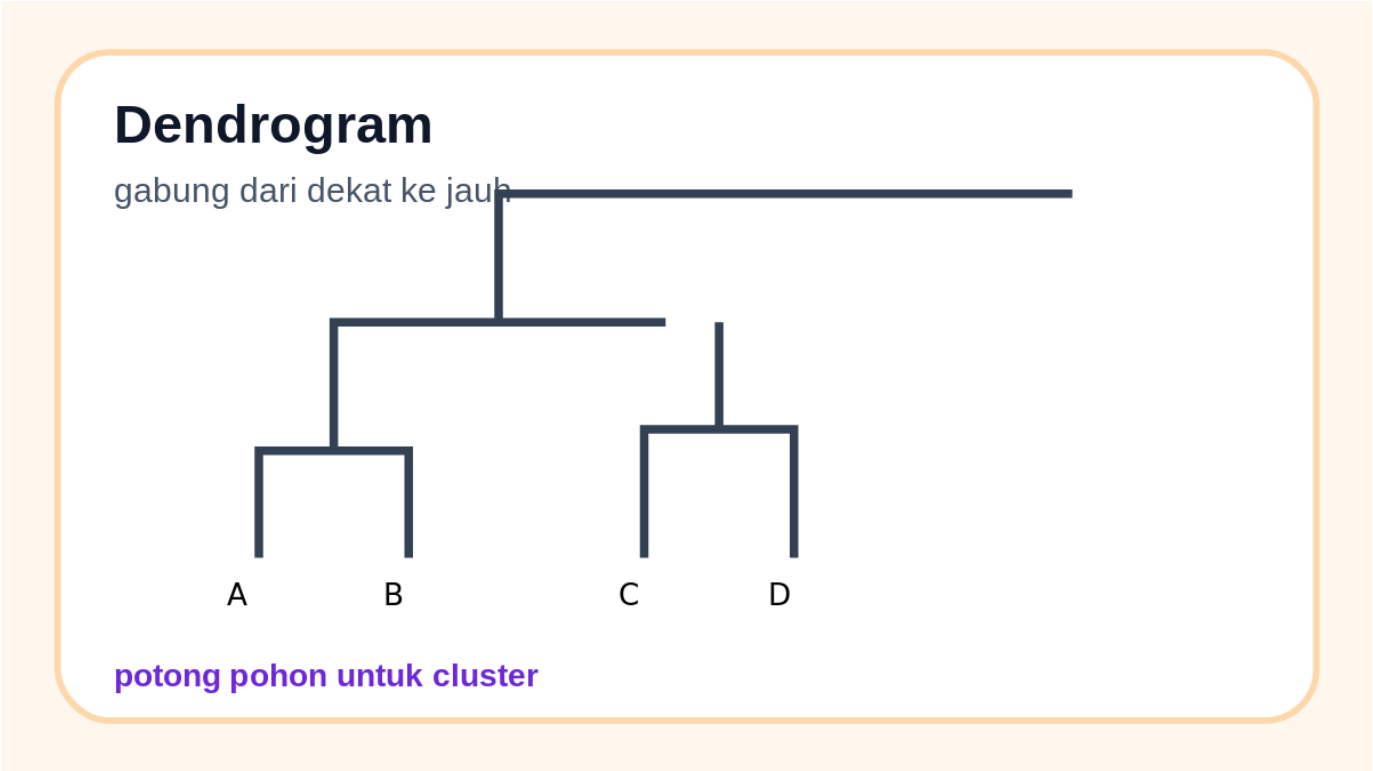
Jarak terdekat pertama A-B:

|1-2| = 1

Jarak C-D:

|8-10| = 2

Maka A dan B bergabung dulu, lalu C dan D, lalu dua cluster besar bergabung.



Dendrogram hierarchical clustering

Cara membaca dendrogram: daun adalah titik awal. Cabang yang bergabung rendah berarti jaraknya dekat. Jika kita memotong dendrogram pada tinggi tertentu, kita mendapatkan jumlah cluster tertentu.

Kapan berguna?

Hierarchical clustering berguna saat kita ingin melihat struktur bertingkat: jenis produk, kelompok dokumen, atau pola pelanggan dari kasar ke rinci. Kelemahannya: untuk data sangat besar bisa mahal secara komputasi.

Tes cepat subbab 13

1. Apa beda hierarchical clustering dan k-means?

2. Apa arti cabang rendah pada dendrogram?

3. Untuk data `[1,2,8,10]`, pasangan mana yang bergabung dulu?

Subbab 14 — DBSCAN: cluster berdasarkan kepadatan

Inti subbab: DBSCAN menemukan area padat dan bisa menandai noise tanpa menentukan jumlah cluster di awal.

DBSCAN memakai dua parameter:

```
eps    = radius tetangga
minPts = jumlah minimum titik dalam radius agar menjadi core point
```

Jenis titik:

```
core point   → punya cukup tetangga dalam eps
border point → dekat core, tetapi tetangganya kurang
noise point  → tidak cukup dekat dengan cluster padat
```

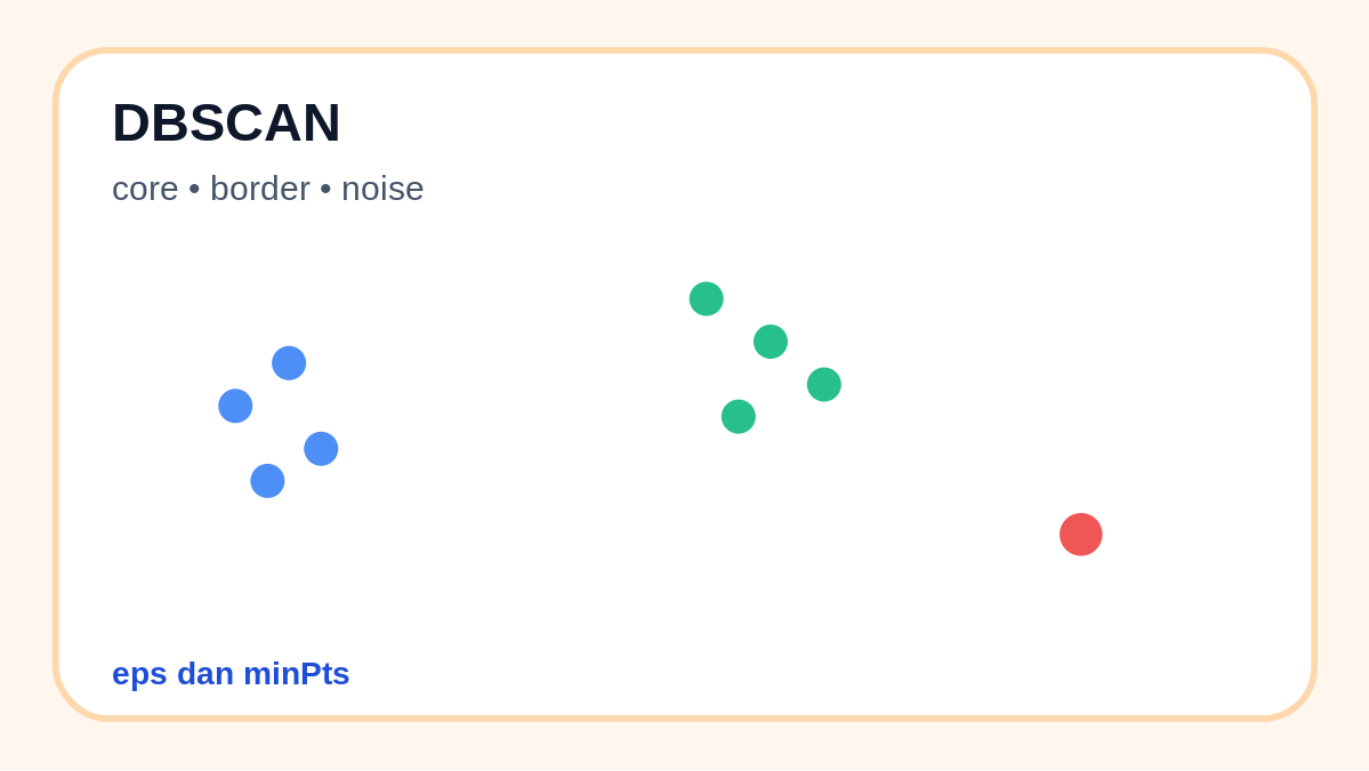
Contoh hitung manual

Data 1D: `[1, 1.2, 1.4, 5, 9]`, `eps=0.5`, `minPts=3` termasuk titik itu sendiri.

Untuk titik 1.2, tetangga dalam radius 0.5:

```
1.0, 1.2, 1.4 → 3 titik
```

Maka 1.2 adalah core point. Titik 5 sendirian, titik 9 sendirian, sehingga keduanya noise jika tidak punya tetangga cukup.



DBSCAN density clustering

Kelebihan: bisa menemukan cluster bentuk tidak bulat dan mendeteksi noise.

Kelemahan: sulit jika kepadatan cluster berbeda-beda. Parameter eps sangat berpengaruh.

Tes cepat subbab 14

1. Apa itu core point?

2. Mengapa DBSCAN tidak perlu memilih k?

3. Untuk data `[1,1.2,1.4,5,9]`, mengapa 5 bisa menjadi noise?

Subbab 15 — Evaluasi tanpa label: inertia, silhouette, dan akal sehat

Inti subbab: tanpa label, evaluasi clustering harus menggabungkan metrik internal dan interpretasi domain.

K-means sering memakai inertia:

$$\text{inertia} = \Sigma_i ||x_i - \mu_{c_i}||^2$$

Inertia makin kecil jika k makin besar, sehingga tidak boleh dipakai sendirian. Silhouette mencoba membandingkan jarak dalam cluster dan jarak ke cluster lain:

$$s = (b - a) / \max(a, b)$$

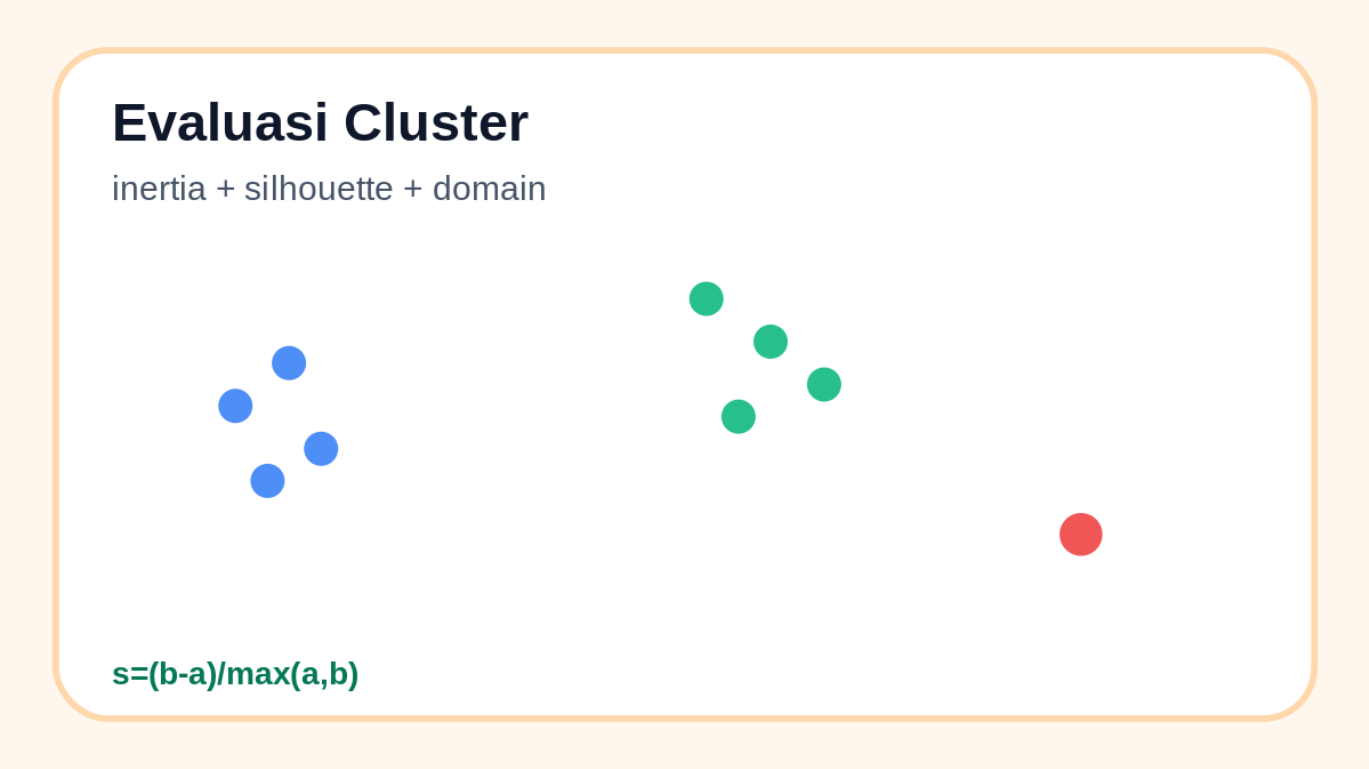
a = rata-rata jarak ke titik dalam cluster sendiri. b = rata-rata jarak ke cluster terdekat lainnya. Nilai mendekati 1 bagus, sekitar 0 ambigu, negatif mencurigakan.

Contoh hitung silhouette mini

Untuk satu titik:

a = 2
b = 5
s = (5-2)/max(2,5) = 3/5 = 0,6

Titik ini cukup cocok dengan clusternya.



Evaluasi clustering

Akal sehat domain

Jika cluster pelanggan menghasilkan strategi yang tidak masuk akal, metrik bagus belum cukup. Tanyakan:

Apakah cluster stabil jika data sedikit berubah?
Apakah cluster bisa dijelaskan?
Apakah cluster membantu keputusan nyata?
Apakah ada risiko diskriminasi atau bias?

Tes cepat subbab 15

1. Mengapa inertia selalu turun saat k bertambah?

2. Hitung silhouette jika a=3 dan b=4.

3. Mengapa metrik internal perlu dilengkapi interpretasi manusia?

Subbab 16 — Dimensionality reduction: merangkum fitur tanpa kehilangan inti

Inti subbab: dimensionality reduction mengubah data berdimensi tinggi menjadi representasi lebih rendah.

Jika data punya 100 fitur, sulit divisualkan dan kadang penuh noise. Dimensionality reduction mencari representasi:

$$z = g(x), \text{ dengan } \dim(z) < \dim(x)$$

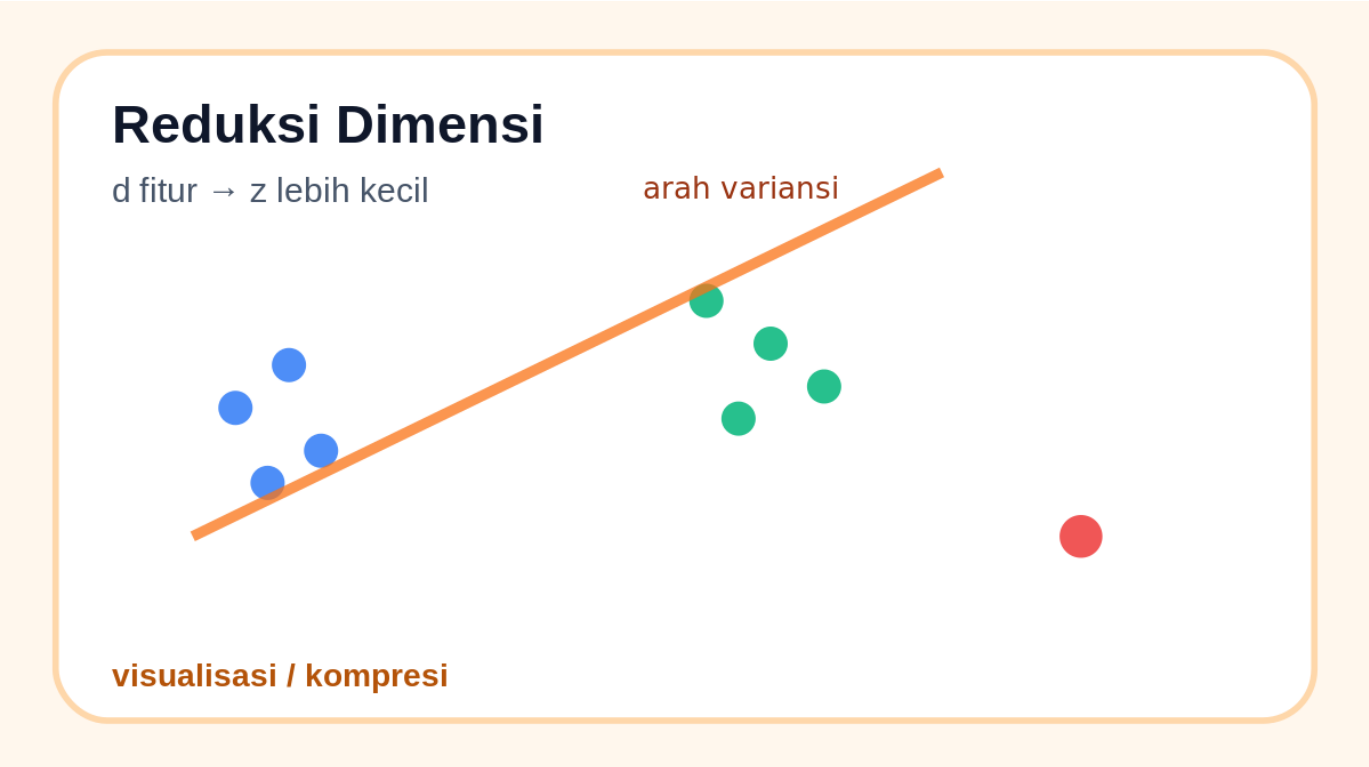
Tujuannya bisa visualisasi, kompresi, denoising, atau mempercepat model berikutnya. PCA adalah metode linear klasik. t-SNE dan UMAP populer untuk visualisasi non-linear. Autoencoder memakai neural network untuk representasi.

Contoh sederhana

Jika dua fitur sangat berkorelasi:

```
luas_rumah dan jumlah_kamar
```

Keduanya mungkin bisa dirangkum menjadi satu arah “ukuran rumah”. PCA mencari arah seperti itu secara matematis.



Dimensionality reduction

Bahaya: visualisasi 2D bisa menipu. Jarak di plot hasil t-SNE/UMAP tidak selalu sama dengan jarak asli. Jangan mengambil keputusan bisnis besar hanya dari gambar cantik.

Tes cepat subbab 16

1. Apa arti `dim(z) < dim(x)`?

2. Mengapa data berdimensi tinggi sulit divisualkan?

3. Sebutkan satu risiko visualisasi reduksi dimensi.

Subbab 17 — PCA: variansi, kovarians, dan arah utama

Inti subbab: PCA mencari arah yang menangkap variansi terbesar.

PCA dimulai dengan data yang sudah di-center:

$$x_{centered} = x - mean(x)$$

Lalu menghitung kovarians:

$$Cov(X) = (1/n)\ X_{centered}^{T}\ X_{centered}$$

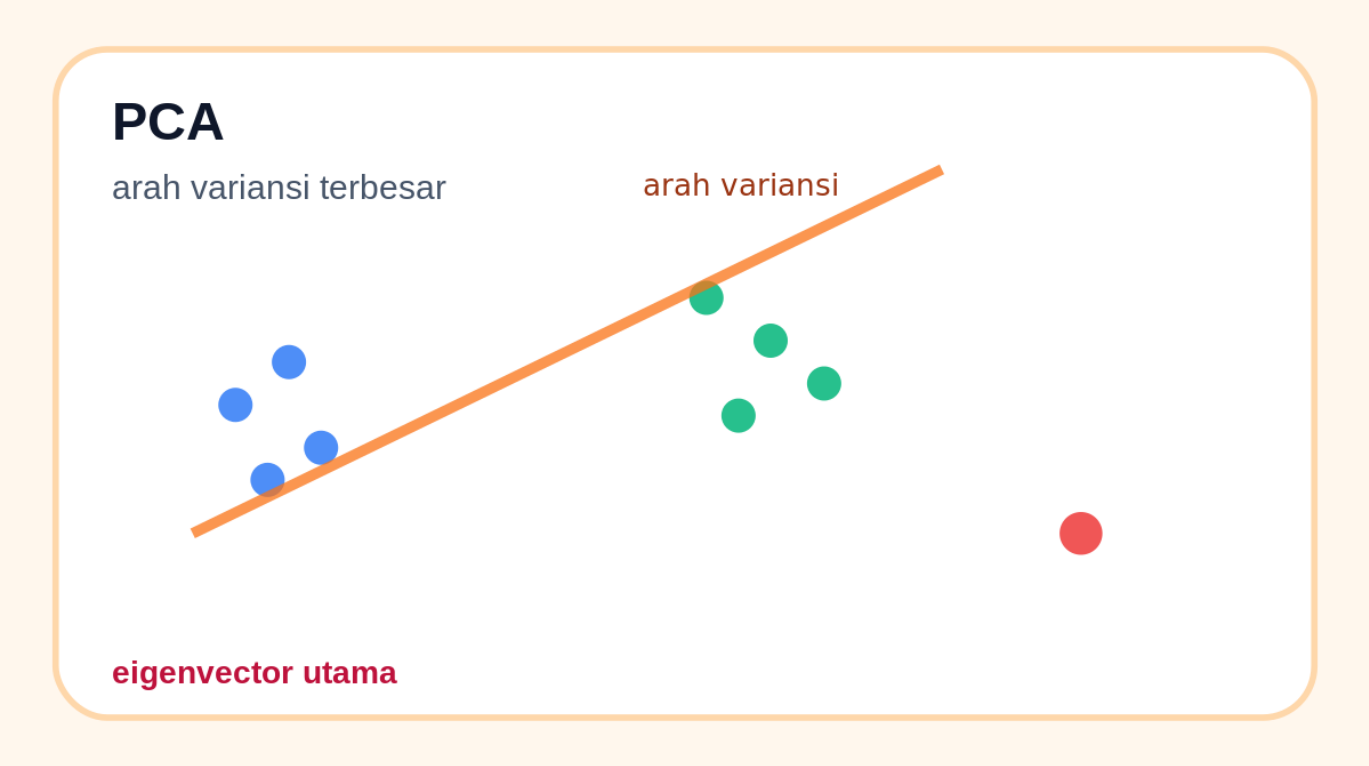
PCA mencari vektor arah w yang memaksimalkan variansi proyeksi:

$$\text{maximize } Var(Xw), \text{ dengan } ||w|| = 1$$

Solusinya adalah eigenvector dari matriks kovarians. Eigenvector dengan eigenvalue terbesar menjadi principal component pertama.

Intuisi visual

Jika titik data membentuk awan memanjang diagonal, arah diagonal menangkap variasi terbesar. PCA memilih arah itu sebagai sumbu baru.



PCA arah variansi

Contoh kovarians kecil

Data 2D yang sudah center:

```
[-1, -1]
[ 0,  0]
[ 1,  1]
```

Variasi berada di arah diagonal [1,1]. PCA akan memilih arah diagonal sebagai komponen utama.

Catatan: PCA linear. Jika struktur data melengkung seperti bulan sabit, PCA mungkin tidak cukup.

Tes cepat subbab 17

1. Mengapa data perlu di-center sebelum PCA?

2. Apa arti eigenvalue besar dalam PCA?

3. Jika awan titik memanjang horizontal, arah principal component pertama kira-kira ke mana?

Subbab 18 — Contoh hitung PCA 2D secara terstruktur

Inti subbab: PCA bisa dihitung manual pada kasus kecil agar rumus tidak terasa magis.

Gunakan data yang sudah di-center:

A = [-1, -1]
B = [0, 0]
C = [1, 1]

Matriks X:

X = [[-1, -1],
    [0, 0],
    [1, 1]]

Kovarians populasi:

Cov = (1/3) $X^T X$

Hitung elemen:

var_x = $((-1)^2 + 0^2 + 1^2)/3 = 2/3$
var_y = $((-1)^2 + 0^2 + 1^2)/3 = 2/3$
cov_xy = $((-1)(-1) + 0 \cdot 0 + 1 \cdot 1)/3 = 2/3$

Jadi:

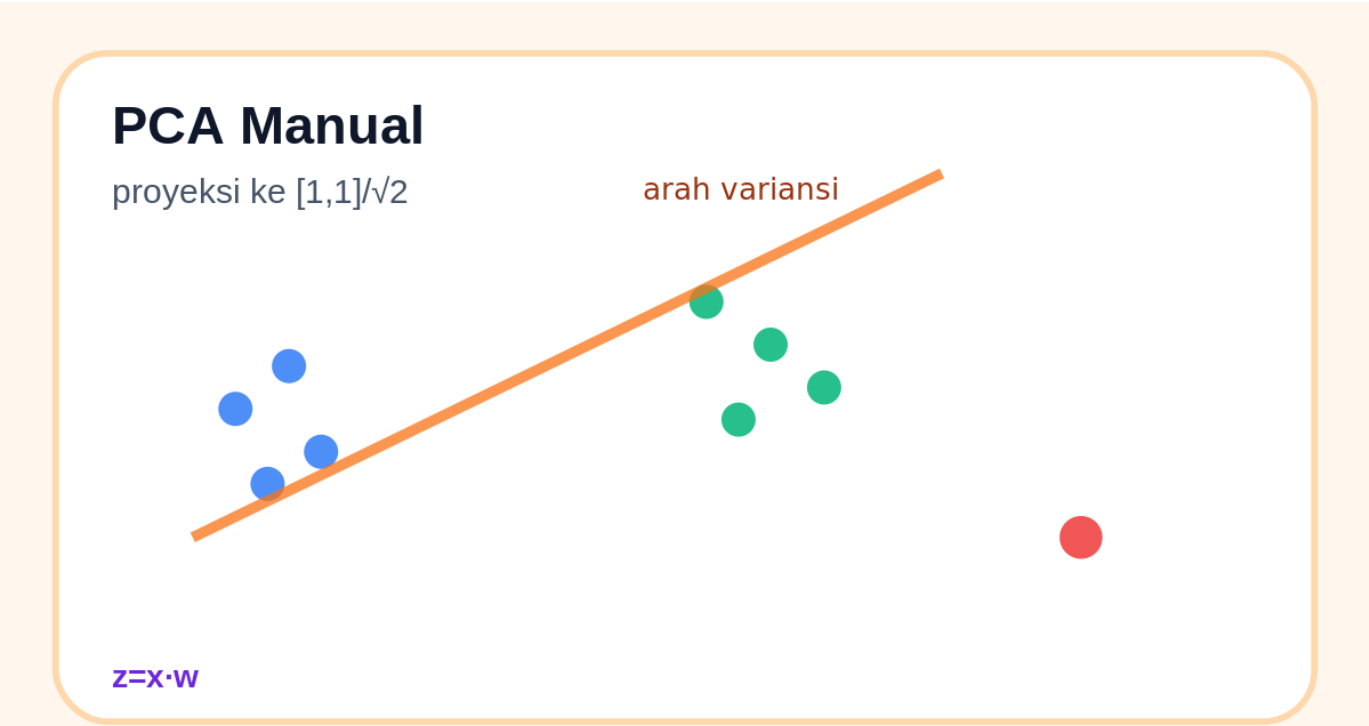
Cov = [[2/3, 2/3],
      [2/3, 2/3]]

Eigenvector utama adalah arah [1,1] yang dinormalisasi:

w_1 = [1/sqrt(2), 1/sqrt(2)]

Proyeksi titik C [1,1] ke w■:

z_C = [1,1] · [1/sqrt(2), 1/sqrt(2)]
   = 2/sqrt(2)
   = sqrt(2)
   ≈ 1,414



PCA contoh hitung

Makna: dua fitur yang bergerak bersama bisa diringkas menjadi satu skor sepanjang diagonal.

Tes cepat subbab 18

1. Hitung proyeksi titik `[-1, -1]` ke w■.

2. Mengapa arah `[1,1]` lebih baik daripada `[1,0]` untuk data ini?

3. Apa yang hilang ketika data 2D diringkas menjadi 1D?

Subbab 19 — Embedding: mengubah objek menjadi vektor bermakna

Inti subbab: embedding adalah representasi vektor untuk objek seperti kata, produk, gambar, atau pengguna.

Embedding menaruh objek ke ruang vektor sehingga objek mirip berada dekat. Dalam NLP, kata “nasi” dan “makan” mungkin dekat. Dalam e-commerce, produk yang sering dibeli bersama bisa dekat.

Secara sederhana:

objek → encoder → vektor h ∈ R^d

Kemiripan embedding sering dihitung dengan cosine similarity:

cos(a,b) = (a·b)/(||a||||b||)

Contoh hitung cosine

a = [1, 1]
b = [2, 2]

Dot product:

a·b = 1·2 + 1·2 = 4

Norma:

$||a|| = \sqrt{1^2+1^2}=\sqrt{2}$
$||b|| = \sqrt{2^2+2^2}=\sqrt{8}=2\sqrt{2}$

Cosine:

$4 / (\sqrt{2}\cdot 2\sqrt{2}) = 4/4 = 1$

Arah sama, jadi similarity maksimum.

Embedding vektor

Catatan etika: embedding belajar dari data. Jika data mengandung bias sosial, embedding bisa ikut membawa bias.

Tes cepat subbab 19

1. Apa itu embedding?

2. Mengapa cosine similarity lebih peduli arah daripada panjang?

3. Hitung cosine untuk `[1,0]` dan `[0,1]`.

Subbab 20 — Representation learning: autoencoder dan self-supervised learning

Inti subbab: representation learning belajar fitur berguna, bukan hanya memakai fitur buatan manusia.

Autoencoder punya dua bagian:

encoder: $x \rightarrow h$
decoder: $h \rightarrow \hat{x}$

Tujuannya meminimalkan reconstruction loss:

$L = ||x - \hat{x}||^2$

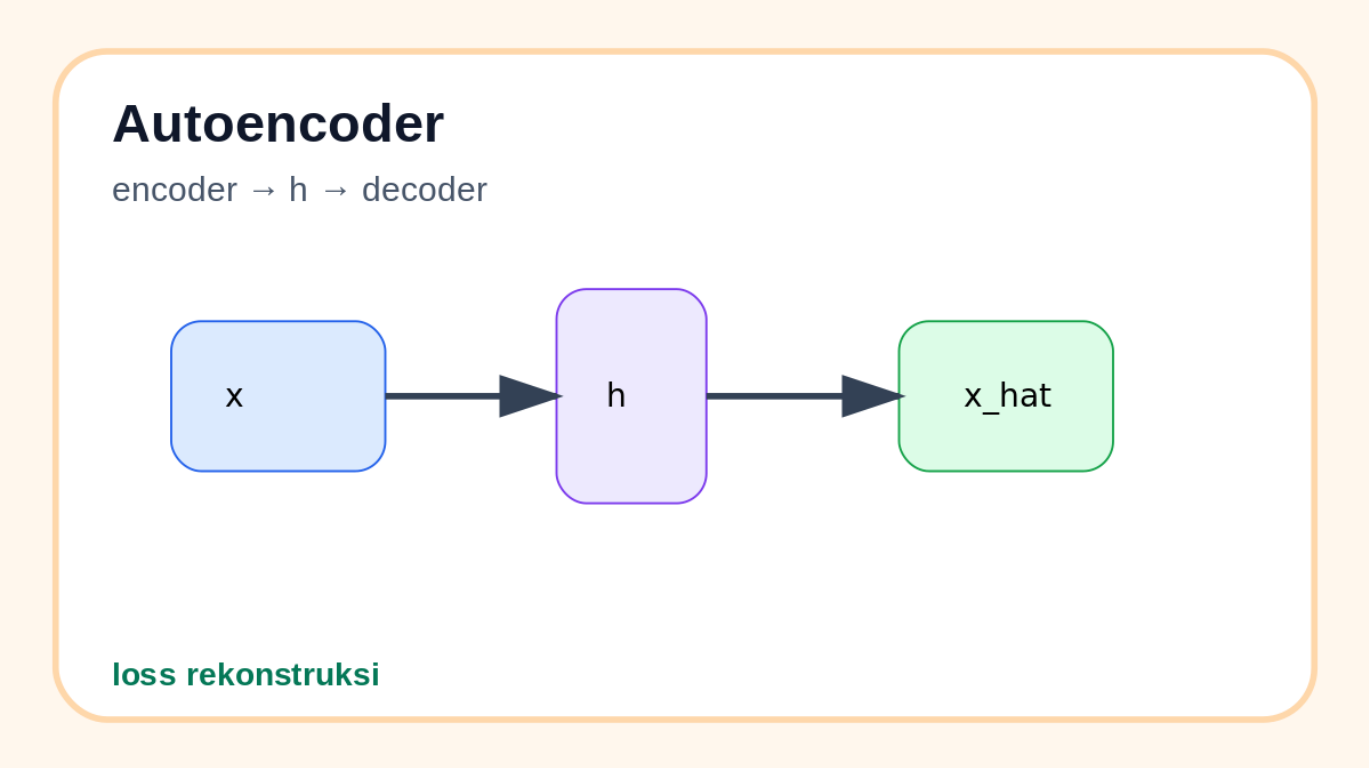
Jika h lebih kecil dari x, model dipaksa menyimpan informasi penting. Ini mirip kompresi, tetapi dipelajari dari data.

Self-supervised learning membuat “label” dari data sendiri. Contoh:

sembunyikan sebagian kata → prediksi kata yang hilang
potong dua augmentasi gambar → buat representasinya dekat
urutkan potongan sinyal → prediksi urutan

Contoh hitung loss rekonstruksi

x = [2, 4, 6]
x_hat = [2, 5, 5]
error = [0, -1, 1]
L = 0^2 + $(-1)^2$ + 1^2 = 2



Autoencoder

Makna: jika loss kecil, representasi h menyimpan cukup informasi untuk membangun ulang input. Tetapi representasi bagus untuk rekonstruksi belum tentu bagus untuk semua tugas.

Tes cepat subbab 20

1. Apa peran encoder dan decoder?

2. Hitung reconstruction loss untuk `x=[1,2]`, `x_hat=[2,2]`.

3. Mengapa self-supervised learning disebut belajar dari data sendiri?

Subbab 21 — Anomaly detection: mencari yang tidak biasa

Inti subbab: anomaly detection memberi skor keanehan, bukan selalu label pasti.

Anomali adalah data yang berbeda dari pola umum. Contoh: transaksi sangat besar, login dari lokasi tidak biasa, sensor mesin tiba-tiba melonjak, atau aktivitas belajar yang turun drastis.

Metode sederhana memakai z-score:

$$z = (x - \mu) / \sigma$$

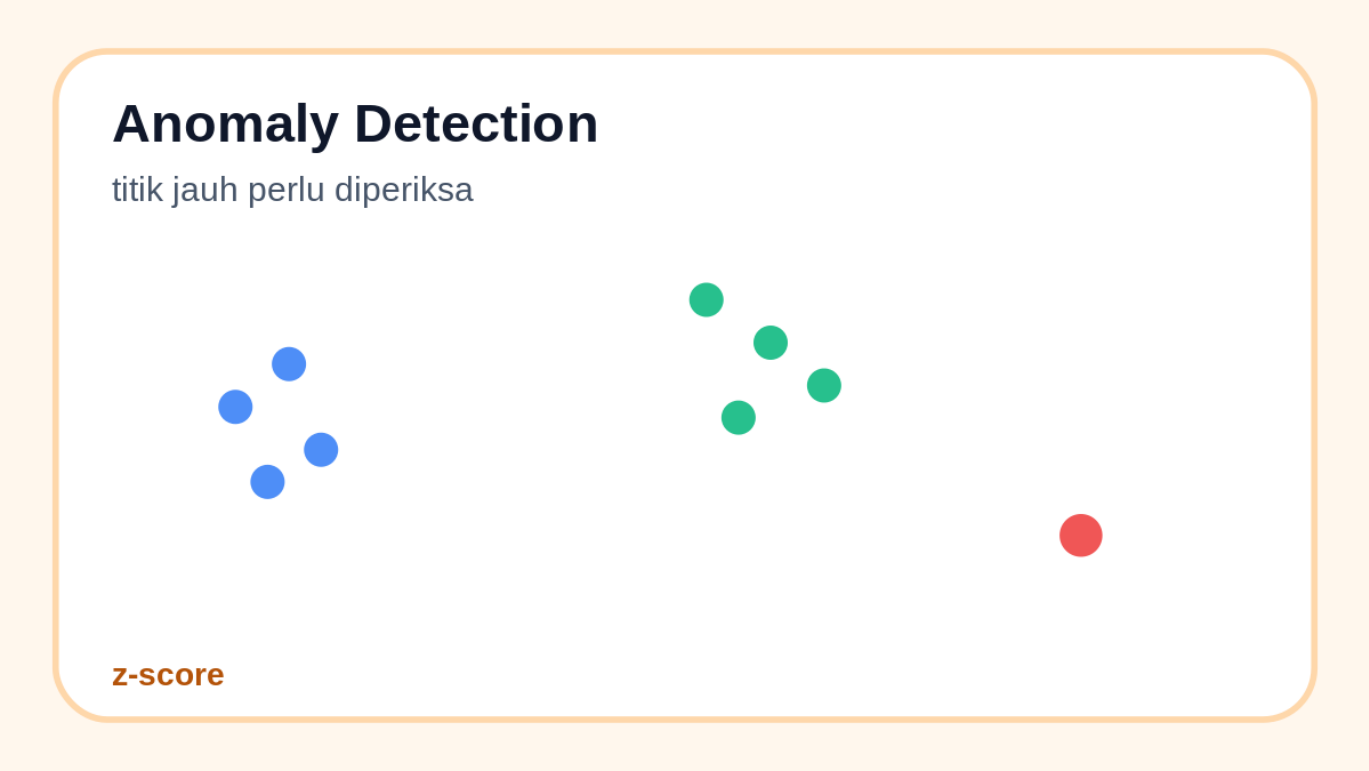
Jika $|z|$ besar, nilai dianggap tidak biasa. Batas umum misalnya $|z| > 3$, tetapi domain menentukan.

Contoh hitung

Data transaksi rata-rata $\mu=100$ ribu, standar deviasi $\sigma=20$ ribu. Transaksi baru $x=170$ ribu.

z = (170 - 100)/20 = 70/20 = 3,5

Karena $z=3,5$, transaksi ini layak diperiksa. Bukan berarti pasti fraud.



Anomaly detection

Kesalahan umum: menganggap semua anomali buruk. Dalam bisnis, anomali bisa pelanggan VIP. Dalam sensor, anomali bisa kerusakan. Dalam kreativitas, anomali bisa ide baru.

Tes cepat subbab 21

1. Hitung z-score untuk $x=130$, $\mu=100$, $\sigma=15$.
2. Mengapa anomali tidak selalu berarti kesalahan?
3. Apa risiko threshold terlalu ketat?

Subbab 22 — Bias, stabilitas, dan kesalahan umum unsupervised learning

Inti subbab: tanpa label bukan berarti tanpa bias.

Unsupervised learning sering terasa objektif karena tidak memakai label manusia. Namun bias tetap bisa masuk lewat data, fitur, scaling, algoritma, dan interpretasi.

Kesalahan umum:

1. Memakai fitur sensitif tanpa pertimbangan etika
2. Tidak melakukan scaling
3. Memilih k karena gambar terlihat bagus saja
4. Memberi nama cluster terlalu stereotip
5. Menganggap visualisasi 2D sebagai realitas penuh
6. Tidak menguji stabilitas terhadap sampel berbeda

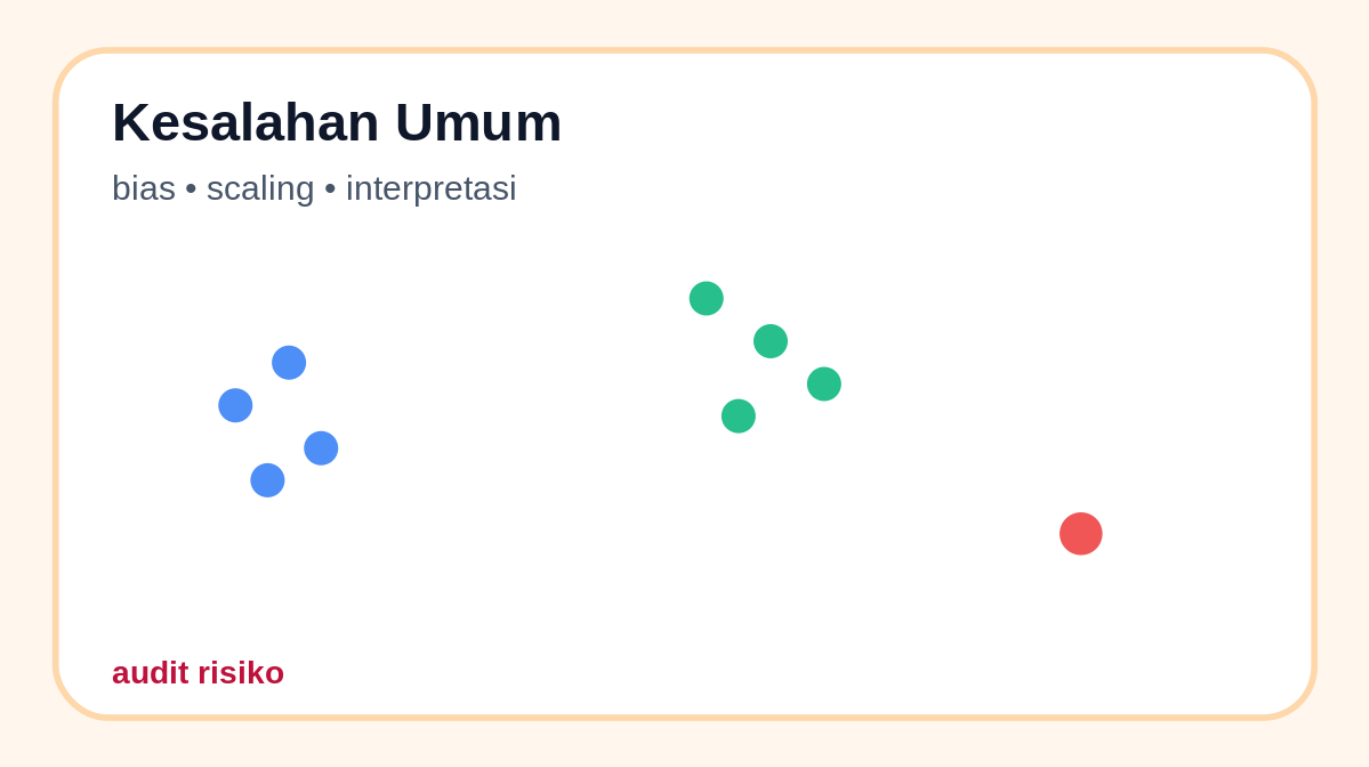
Contoh stabilitas sederhana

Jika k-means pada data penuh memberi 3 cluster, lalu pada 80% sampel acak cluster berubah total, segmentasi belum stabil. Stabilitas bisa diuji dengan menjalankan algoritma beberapa kali dan membandingkan hasil.

Skor risiko interpretasi

```
risk = dampak_keputusan × ketidakpastian_cluster
```

Jika cluster dipakai hanya untuk eksplorasi konten, risiko rendah. Jika dipakai untuk menentukan akses pinjaman, risiko tinggi dan butuh validasi ketat.



Kesalahan umum

Tes cepat subbab 22

1. Mengapa data tanpa label tetap bisa bias?
2. Sebutkan dua kesalahan umum clustering.
3. Mengapa cluster untuk keputusan berisiko tinggi perlu audit tambahan?

Subbab 23 — Memilih metode: peta praktis unsupervised learning

Inti subbab: pilih metode berdasarkan tujuan, bentuk data, ukuran data, dan kebutuhan penjelasan.

Peta praktis:

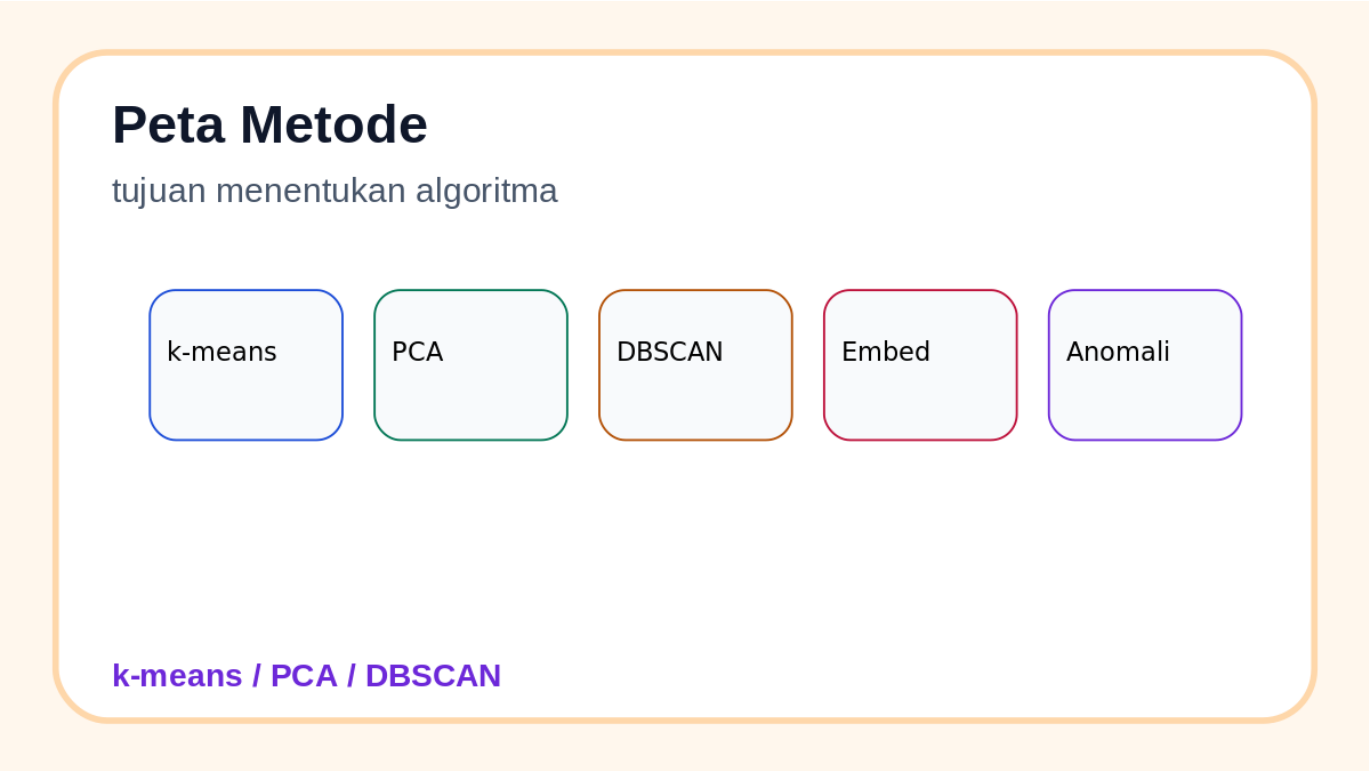
```
Ingin segmentasi sederhana dan cluster bulat? → k-means
Ingin struktur bertingkat? → hierarchical clustering
Ingin cluster bentuk aneh dan noise? → DBSCAN
Ingin visualisasi/kompresi linear? → PCA
Ingin embedding objek kompleks? → representation learning
Ingin titik tidak biasa? → anomaly detection
```

Contoh keputusan

Kasus 1: UMKM ingin segmentasi pelanggan dari 5 fitur numerik, butuh penjelasan mudah. Mulai dari scaling + k-means + interpretasi centroid.

Kasus 2: Peneliti ingin melihat hubungan dokumen berita. Gunakan embedding teks + visualisasi 2D, lalu cek manual topik.

Kasus 3: Aplikasi pembayaran ingin memeriksa transaksi aneh. Mulai dari aturan/z-score baseline, lalu pertimbangkan isolation/anomaly model.



Peta pemilihan metode

Pertanyaan sebelum memilih metode

Apa tujuan keputusan?
Apa fitur yang tersedia?
Apakah skala fitur sebanding?
Apakah hasil harus mudah dijelaskan?
Berapa biaya salah interpretasi?

Tes cepat subbab 23

1. Kapan k-means cocok?

2. Kapan DBSCAN lebih menarik daripada k-means?

3. Mengapa embedding perlu evaluasi domain?

Subbab 24 — Praktikum Bab 8: membangun peta pelanggan tanpa label

Inti subbab: pembaca akan membuat pipeline unsupervised kecil dari nol memakai Python standard library.

Praktikum Bab 8 memakai dataset pelanggan mini. Targetnya bukan mengejar library canggih, tetapi memahami langkah-langkah:

1. Siapkan data pelanggan
2. Standardisasi fitur
3. Hitung jarak
4. Jalankan k-means sederhana
5. Hitung inertia
6. Hitung PCA 2D mini
7. Hitung z-score anomali
8. Interpretasi hasil sebagai cerita bisnis

File praktikum:

chapters/08-unsupervised-representation-learning/code/unsupervised_playground.py
chapters/08-unsupervised-representation-learning/code/unsupervised_playground.ipynb

Contoh interpretasi output

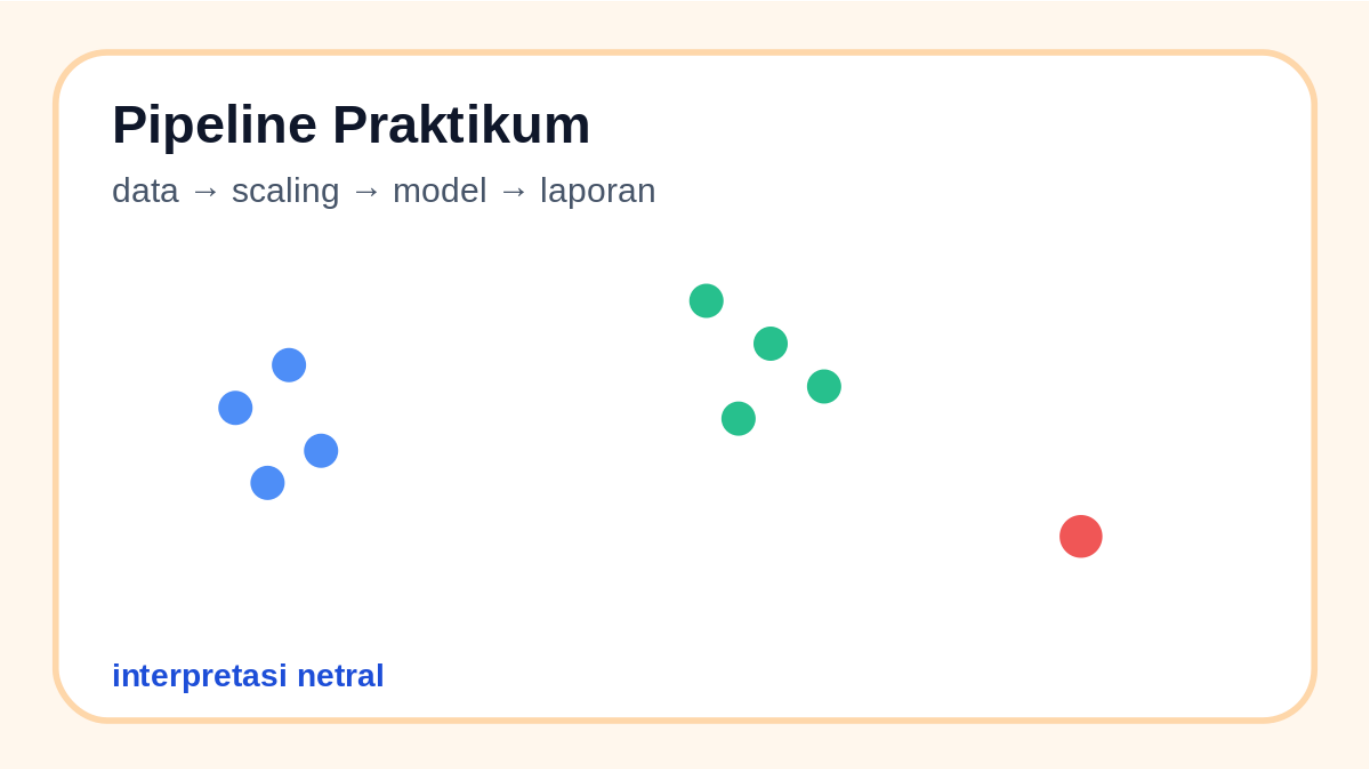
Jika cluster 0 punya rata-rata kunjungan tinggi dan belanja sedang, jangan menamainya “orang baik”. Nama yang lebih aman:

pelanggan rutin bernilai sedang

Jika cluster 1 punya belanja tinggi tetapi kunjungan rendah:

pembeli sesekali bernilai tinggi

Nama cluster harus deskriptif, bukan menghakimi.



Pipeline praktikum

Latihan hitung manual wajib

1. Hitung jarak Euclidean antara pelanggan A `[10,50]` dan B `[11,52]`.
2. Jalankan satu iterasi k-means 1D untuk `[2,4,10,12]`.
3. Hitung inertia dari cluster `[2,4]` dengan centroid 3 dan `[10,12]` dengan centroid 11.
4. Hitung silhouette untuk `a=2`, `b=5`.
5. Hitung z-score transaksi `170` jika `μ=100`, `σ=20`.
6. Hitung proyeksi `[1,1]` ke `[1/sqrt(2), 1/sqrt(2)]`.

Tes cepat subbab 24

1. Mengapa praktikum memakai baseline sederhana dari nol?
2. Apa yang harus ditulis setelah model memberi cluster?
3. Mengapa nama cluster harus netral dan deskriptif?

Pendalaman teknis sebelum praktikum

Bagian ini merangkum rumus yang akan sering muncul saat praktik.

Euclidean distance:
$d(a,b) = \sqrt{\Sigma_j (a_j - b_j)^2}$

K-means objective:
$J = \Sigma_i \|x_i - \mu_{c_i}\|^2$

Centroid:
$\mu = (\Sigma_i x_i)/n$

Standardization:
$z = (x - \mu)/\sigma$

Silhouette:
$s = (b-a)/\max(a,b)$

PCA projection:
$z = x \cdot w$

Reconstruction loss:
$L = \|x - \hat{x}\|^2$

Urutan berpikir yang aman:

fitur → scaling → jarak/representasi → model → metrik → interpretasi → audit risiko

Kamus cara membaca persamaan Bab 8

Bagian ini dibuat khusus agar pembaca tidak hanya melihat simbol, tetapi tahu cara membacanya.

tur.”	Jarak Manhattan; cocok untuk beberapa kasus grid/biaya langkah.			
			b	
	²`	“Jumlah jarak kuadrat tiap titik ke centroid clusternya.”	Objective k-means yang ingin diperkecil.	

	²`	“Loss adalah jarak kuadrat input dan rekonstruksi.”	Objective autoencoder.	

Jika pembaca lupa simbol, kembali ke tabel ini sebelum melanjutkan latihan.

Praktikum terpadu Bab 8

1. Jalankan `unsupervised_playground.py` dari terminal atau VS Code.

2. Buka notebook `.ipynb` di Jupyter, Colab, atau Kaggle.

3. Buka folder `code/outputs/` dan baca empat plot SVG: `histogram_spend.svg`, `scatter_clusters.svg`, `linear_regression_residuals.svg`, dan `anomaly_zscore.svg`.

4. Ubah data pelanggan: tambahkan 2 pelanggan baru.

5. Ubah nilai missing/outlier, lalu catat bagaimana log cleansing berubah.

6. Ubah `k` dari 2 menjadi 3 dan catat perubahan interpretasi.

7. Ubah fitur yang dipakai, lalu cek apakah cluster berubah.

8. Tulis laporan mini:

- fitur yang digunakan, - alasan scaling, - log cleansing, - plot yang dibaca, - slope/intercept regresi linear, - residual besar, - inertia, - interpretasi centroid, - titik anomali, - risiko salah tafsir.

Pembahasan latihan hitung terstruktur Bab 8

Bagian ini sengaja dibuat seperti buku kerja. Pembaca dianjurkan menutup jawaban terlebih dahulu, menghitung di kertas, lalu mencocokkan langkahnya. Tujuannya bukan menghafal rumus, tetapi melihat pola berulang: definisikan data, pilih rumus, hitung pelan-pelan, tafsirkan hasil.

Latihan 1 — Jarak Euclidean pelanggan

Diberikan:

A = [10, 50]
B = [11, 52]
C = [2, 15]

Jarak A ke B:

d(A,B) = sqrt((10-11)² + (50-52)²)
= sqrt((-1)² + (-2)²)
= sqrt(1 + 4)
= sqrt(5)
≈ 2,24

Jarak A ke C:

d(A,C) = sqrt((10-2)² + (50-15)²)
= sqrt(8² + 35²)
= sqrt(64 + 1225)
= sqrt(1289)
≈ 35,90

Kesimpulan: A jauh lebih dekat ke B. Jika fitur benar-benar relevan, A dan B mungkin perilakunya mirip. Tetapi kata “jika” penting: jarak hanya bermakna jika fitur dan scaling masuk akal.

Latihan 2 — Satu iterasi k-means 1D

Diberikan data:

x = [2, 4, 10, 12]
centroid awal: $\mu_1 = 2$, $\mu_2 = 12$

Assign titik ke centroid terdekat:

x=2 → jarak ke μ_1=0, ke μ_2=10 → cluster 1
x=4 → jarak ke μ_1=2, ke μ_2=8 → cluster 1
x=10 → jarak ke μ_1=8, ke μ_2=2 → cluster 2
x=12 → jarak ke μ_1=10, ke μ_2=0 → cluster 2

Update centroid:

μ_1 baru = (2+4)/2 = 3
μ_2 baru = (10+12)/2 = 11

Hitung inertia:

$J = (2-3)^2 + (4-3)^2 + (10-11)^2 + (12-11)^2$
$= 1 + 1 + 1 + 1$
$= 4$

Kesimpulan: satu iterasi sudah membuat pusat cluster bergerak dari titik ekstrem ke rata-rata kelompok.

Latihan 3 — Silhouette sederhana

Diberikan:

a = 2 # rata-rata jarak ke cluster sendiri
b = 5 # rata-rata jarak ke cluster lain terdekat

Rumus:

$s = (b - a) / \max(a, b)$

Hitung:

$s = (5 - 2) / \max(2, 5)$
$= 3 / 5$
$= 0,6$

Interpretasi: nilai 0,6 cukup baik karena titik lebih dekat ke cluster sendiri daripada cluster lain. Jika s mendekati 0, titik berada di perbatasan. Jika negatif, titik mungkin lebih cocok di cluster lain.

Latihan 4 — PCA manual pada arah diagonal

Diberikan titik yang sudah di-center:

[-1, -1], [0, 0], [1, 1]

Arah utama:

$w = [1/\sqrt{2}, 1/\sqrt{2}]$

Proyeksi titik `[1,1]`:

$z = [1,1] \cdot [1/\sqrt{2}, 1/\sqrt{2}]$
$= 1/\sqrt{2} + 1/\sqrt{2}$
$= 2/\sqrt{2}$
$= \sqrt{2}$
$\approx 1,414$

Proyeksi titik `[-1,-1]`:

$z = [-1,-1] \cdot [1/\sqrt{2}, 1/\sqrt{2}]$
$= -1/\sqrt{2} - 1/\sqrt{2}$
$= -\sqrt{2}$
$\approx -1,414$

Interpretasi: data 2D yang bergerak sepanjang diagonal dapat diringkas menjadi satu angka posisi di sepanjang diagonal.

Latihan 5 — Cosine similarity embedding

Diberikan:

```
a = [1, 0]
b = [0, 1]
```

Dot product:

```
a·b = 1·0 + 0·1 = 0
```

Norma:

```
||a|| = 1
||b|| = 1
```

Cosine:

```
cos(a,b) = 0/(1·1) = 0
```

Interpretasi: dua vektor tegak lurus tidak mirip secara arah. Dalam embedding, nilai 0 bukan selalu berarti “bermusuhan”; ia berarti arah representasinya tidak sejalan.

Latihan 6 — Z-score anomali

Diberikan rata-rata transaksi $\mu=100$, standar deviasi $\sigma=20$, transaksi baru $x=170$.

```
z = (x - μ) / σ
  = (170 - 100) / 20
  = 70 / 20
  = 3,5
```

Interpretasi: transaksi ini cukup jauh dari rata-rata. Tindakan yang benar bukan langsung menolak transaksi, tetapi memberi tanda untuk pemeriksaan tambahan. Bisa jadi fraud, bisa jadi pelanggan membeli untuk acara besar.

Studi kasus mini — segmentasi pelanggan warung kopi

Bayangkan pemilik warung kopi ingin memahami pelanggan tanpa label. Ia memakai tiga fitur:

```
kunjungan per bulan
rata-rata belanja
persen pesanan kopi susu
```

Langkah aman:

1. Bersihkan data yang jelas salah.
2. Standardisasi fitur.
3. Jalankan k-means dengan beberapa nilai k.
4. Catat inertia dan interpretasi centroid.
5. Periksa apakah cluster stabil saat data diacak sedikit.
6. Beri nama cluster secara netral.
7. Jangan memakai cluster untuk keputusan sensitif tanpa audit.

Contoh interpretasi centroid:

```
centroid A = kunjungan tinggi, belanja sedang, kopi susu tinggi
nama netral: pelanggan rutin kopi susu

centroid B = kunjungan rendah, belanja tinggi, kopi susu rendah
nama netral: pembeli sesekali bernilai tinggi
```

Nama yang buruk:

```
pelanggan malas
pelanggan miskin
```

pelanggan tidak penting

Nama seperti itu membawa bias dan tidak layak untuk produk komersial yang bertanggung jawab.

Checklist debugging Bab 8

Jika hasil unsupervised learning terasa aneh, periksa:

Apakah data punya duplikasi ekstrem?
Apakah fitur sudah diskalakan?
Apakah ada fitur yang sebenarnya ID atau kode kategori palsu?
Apakah outlier menarik centroid terlalu jauh?
Apakah jumlah cluster dipilih karena alasan bisnis atau hanya karena warna bagus?
Apakah interpretasi cluster divalidasi dengan contoh nyata?

Checklist ini membuat pembaca sadar bahwa unsupervised learning adalah proses investigasi. Model memberi peta awal; manusia tetap bertanggung jawab membaca peta itu.

Catatan teori lanjut — mengapa unsupervised learning sering sulit tetapi penting

Unsupervised learning lebih sulit dievaluasi daripada supervised learning karena tidak ada jawaban benar yang langsung tersedia. Pada supervised learning, kita bisa menghitung accuracy, precision, recall, MAE, atau MSE terhadap label. Pada unsupervised learning, kita sering hanya punya sinyal tidak langsung: cluster terlihat rapat, inertia menurun, silhouette membaik, atau representasi berguna untuk tugas berikutnya. Karena itu, kemampuan teknis harus dipasangkan dengan pertanyaan domain.

Pertanyaan pertama: struktur apa yang sedang dicari? Jika kita mencari segmen pelanggan, cluster harus membantu strategi layanan. Jika kita mencari representasi dokumen, embedding harus membuat dokumen bertopik mirip berada dekat. Jika kita mencari anomali transaksi, skor anomali harus membantu investigasi tanpa membanjiri tim dengan false alarm.

Pertanyaan kedua: fitur apa yang mendefinisikan kemiripan? Dua orang bisa mirip dari sisi belanja, tetapi berbeda dari sisi waktu kunjungan. Dua dokumen bisa mirip dari sisi kata, tetapi berbeda dari sisi maksud. Dua gambar bisa mirip secara warna, tetapi berbeda secara objek. Unsupervised learning tidak tahu “kemiripan yang benar” kecuali kita mendesain representasi yang tepat.

Pertanyaan ketiga: apakah struktur stabil? Jika sedikit perubahan data membuat cluster berubah total, hasilnya belum layak dijadikan dasar keputusan besar. Stabilitas bisa dicek dengan menjalankan model beberapa kali, mengubah seed, mengambil sampel berbeda, atau membandingkan periode waktu berbeda. Hasil yang stabil tidak otomatis benar, tetapi hasil yang sangat tidak stabil perlu dicurigai.

Pertanyaan keempat: apakah hasil bisa dijelaskan? K-means relatif mudah dijelaskan lewat centroid. Hierarchical clustering bisa dijelaskan lewat dendrogram. PCA bisa dijelaskan lewat arah variansi. Embedding deep learning sering lebih kuat tetapi lebih sulit dijelaskan. Dalam produk komersial, trade-off ini penting: metode yang lebih canggih belum tentu lebih cocok jika tim tidak bisa menjelaskan hasilnya.

Pertanyaan kelima: apa biaya salah tafsir? Cluster untuk eksplorasi konten risiko rendah. Cluster untuk menentukan harga, kredit, beasiswa, atau akses layanan risiko tinggi. Semakin tinggi dampak keputusan, semakin besar kebutuhan audit, dokumentasi, dan validasi manusia.

Menghubungkan Bab 8 ke Bab 9 dan 10

Bab 8 menjadi jembatan menuju neural networks dan deep learning. Embedding, autoencoder, dan self-supervised learning menunjukkan bahwa model modern tidak hanya belajar memprediksi label, tetapi juga belajar representasi. Ketika nanti pembaca melihat neural network, ia sudah mengenal pola:

input x → representasi h → output / rekonstruksi / aksi

Pada supervised learning, h dipakai untuk memprediksi y. Pada autoencoder, h dipakai untuk membangun ulang x. Pada self-supervised learning, h dilatih dari tugas buatan seperti menebak bagian data yang disembunyikan. Pada generative AI, representasi internal membantu model memahami konteks dan menghasilkan token berikutnya.

Contoh analogi: peta kota dan peta data

Bayangkan kita punya peta kota. Jarak fisik membantu, tetapi tidak selalu cukup. Dua tempat dekat secara geografis bisa berbeda fungsi: rumah sakit dan pasar. Dua tempat jauh bisa mirip fungsi: dua kampus di kota berbeda. Begitu juga data. Jarak Euclidean pada fitur mentah hanyalah satu jenis peta. Embedding mencoba membuat peta yang lebih bermakna: objek yang fungsi atau konteksnya mirip diletakkan dekat.

Karena itu, unsupervised learning sering dimulai dari pertanyaan: peta seperti apa yang kita butuhkan? Peta untuk navigasi? Peta untuk segmentasi? Peta untuk deteksi risiko? Peta untuk kompresi? Metode yang dipilih mengikuti peta yang diinginkan.

Mini-rubrik kedalaman pemahaman

Pembaca dianggap memahami Bab 8 jika dapat melakukan enam hal berikut:

1. Menjelaskan perbedaan data berlabel dan tanpa label.
2. Menghitung jarak dan menjelaskan efek scaling.
3. Menjalankan k-means manual minimal satu iterasi.
4. Membaca dendrogram dan plot PCA secara kritis.
5. Menghitung z-score dan tidak langsung memvonis anomali.
6. Menulis interpretasi cluster yang netral, berbasis fitur, dan sadar risiko.

Jika keenam hal ini sudah bisa dilakukan, pembaca siap masuk ke Bab 9 tentang neural networks dengan fondasi representasi yang lebih kuat. Pembaca juga sudah memiliki kebiasaan penting: selalu menghubungkan angka, gambar, rumus, kode, dan dampak keputusan nyata.

Ringkasan Bab 8

• Unsupervised learning bekerja tanpa label target.

• Data biasanya ditulis sebagai matriks $X \in R^{n \times d}$.

• Jarak dan kemiripan menentukan arti “mirip”.

• Scaling penting agar fitur besar tidak mendominasi.

• K-means mencari centroid yang meminimalkan jarak kuadrat.

• Hierarchical clustering membangun dendrogram bertingkat.

• DBSCAN mencari area padat dan dapat menandai noise.

• Evaluasi tanpa label harus memakai metrik dan interpretasi domain.

• PCA mencari arah variansi terbesar.

• Embedding dan representation learning mengubah objek menjadi vektor bermakna.

• Anomaly detection memberi skor keanehan, bukan vonis otomatis.

• Hasil cluster harus diaudit secara etis dan praktis.

Referensi utama bab

• Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space.

• Hotelling, H. (1933). Analysis of a Complex of Statistical Variables into Principal Components.

• MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations.

• Ester, M., Kriegel, H.-P., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

• Bishop, C. M. Pattern Recognition and Machine Learning.

• Murphy, K. P. Machine Learning: A Probabilistic Perspective.

• Goodfellow, Bengio, Courville. Deep Learning.

• scikit-learn documentation: clustering, decomposition, preprocessing.

Catatan validasi internal v0.1

• Draft memakai format subbab seperti Bab 7.

• Setiap subbab punya tes cepat.

• Rumus utama disertai contoh hitung manual.

• Praktikum dirancang berjalan dengan Python standard library agar mudah dijalankan di lokal, VS Code, Jupyter, Colab, dan Kaggle.