

CPMK 09.1

Penerapan Unsupervised Learning pada Dataset

Praktik dengan Python

Discovery Learning · Praktik Terbimbing

IF261220 3 SKS · Program Studi Informatika, Fakultas Ilmu Komputer

UPN "Veteran" Jawa Timur

PERTEMUAN

07

SEBELUM MULAI

Recap Pertemuan 6

Tiga kategori unsupervised

Clustering, Association Rule, dan Anomaly Detection.

K-Means dan penentuan k

Cara kerja iteratif, metode elbow, dan cuplikan kode scikit-learn.

AGENDA PERTEMUAN 7

- | | | | |
|-----------|--------------------------------------|-----------|--------------------------------------|
| 01 | Alur kerja unsupervised pada dataset | 02 | Studi kasus: segmentasi pelanggan |
| 03 | Preprocessing dan memilih k | 04 | Melatih, memvisualisasi, menafsirkan |
| 05 | DBSCAN dan Apriori pada data lain | 06 | Praktik mandiri dan persiapan UTS |

Enam Langkah Menerapkan Unsupervised Learning

Urutan yang sama berlaku untuk hampir semua dataset, apa pun algoritmanya.



Penting: langkah 3 dan 6 paling sering dilupakan. Tanpa preprocessing hasil bisa menyesatkan, tanpa tafsir hasilnya hanya angka.

Segmentasi Pelanggan Toko Ritel

Tujuan: mengelompokkan pelanggan tanpa label agar tim pemasaran bisa menyusun promosi yang tepat.

ID	Usia	Pendapatan (juta Rp/bln)	Skor Belanja (1–100)
1	23	6	82
2	45	18	35
3	31	9	90
4	52	22	18
5	28	14	55
6	39	25	78
7	60	8	12
8	34	12	60

Contoh 8 dari ±200 baris. Nilai bersifat ilustrasi.

YANG PERLU DIPAHAMI

Tidak ada kolom label

Tidak ada kolom "segmen". Model yang harus menemukannya.

Fitur yang dipakai

Pendapatan dan Skor Belanja. Usia disimpan untuk tafsir.

Hasil yang diharapkan

Beberapa segmen dengan perilaku belanja berbeda.

Mengenali Data Sebelum Memodelkan

Selalu mulai dengan melihat data apa adanya.

```
import pandas as pd

df = pd.read_csv("pelanggan.csv")
print(df.head())      # lima baris pertama
print(df.info())      # tipe data dan nilai kosong
print(df.describe())  # min, maks, rata-rata, sebaran
```

Nilai kosong?

Isi atau buang baris yang kosong.

Rentang nilai?

Lihat min dan maks tiap kolom.

Ada nilai aneh?

Pendapatan negatif, usia 0, dan sejenisnya.

LANGKAH 3 · PREPROCESSING

Samakan Skala Fitur

K-Means mengukur jarak, jadi fitur berskala besar akan mendominasi.

PENDAPATAN 6–25 vs SKOR 12–90

Selisih skor belanja (± 78) jauh lebih besar daripada selisih pendapatan (± 19). Tanpa scaling, jarak antar-pelanggan hampir hanya ditentukan oleh skor belanja.

```
from sklearn.preprocessing import StandardScaler  
  
X = df[["Pendapatan", "SkorBelanja"]]  
X_scaled = StandardScaler().fit_transform(X)
```

Hasil: tiap fitur punya rata-rata 0 dan simpangan baku 1, sehingga keduanya berkontribusi seimbang pada jarak.

Memilih k dengan Elbow dan Silhouette

Jalankan K-Means untuk beberapa nilai k, lalu bandingkan dua ukuran.

k	Inertia (makin kecil makin rapat)	Silhouette
2	410	0.41
3	265	0.47
4	170	0.55
5	150	0.50
6	138	0.46

Nilai ilustrasi untuk dataset contoh.

Elbow

Inertia turun tajam sampai k = 4, lalu melandai. Sikunya di k = 4.

Silhouette

Tertinggi di k = 4 (0.55). Dua ukuran sepakat.

Putusan

Pilih k = 4, lalu cek apakah segmennya masuk akal secara bisnis.

Perulangan untuk Membandingkan k

Satu loop menghasilkan angka untuk kedua ukuran.

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

for k in range(2, 7):
    km = KMeans(n_clusters=k, random_state=42)
    labels = km.fit_predict(X_scaled)
    print(k, km.inertia_, silhouette_score(X_scaled, labels))
```

Catatan: random_state=42 membuat hasil bisa diulang. Tanpanya, posisi centroid awal acak dan angka bisa sedikit berbeda tiap dijalankan.

Melatih K-Means dengan $k = 4$

Simpan label klaster kembali ke DataFrame agar mudah dianalisis.

```
km = KMeans(n_clusters=4, random_state=42)
df["klaster"] = km.fit_predict(X_scaled)

print(df["klaster"].value_counts())      # jumlah anggota tiap klaster
print(df.groupby("klaster").mean())      # profil rata-rata tiap klaster
```

fit_predict

Melatih model sekaligus memberi label klaster 0–3.

groupby().mean()

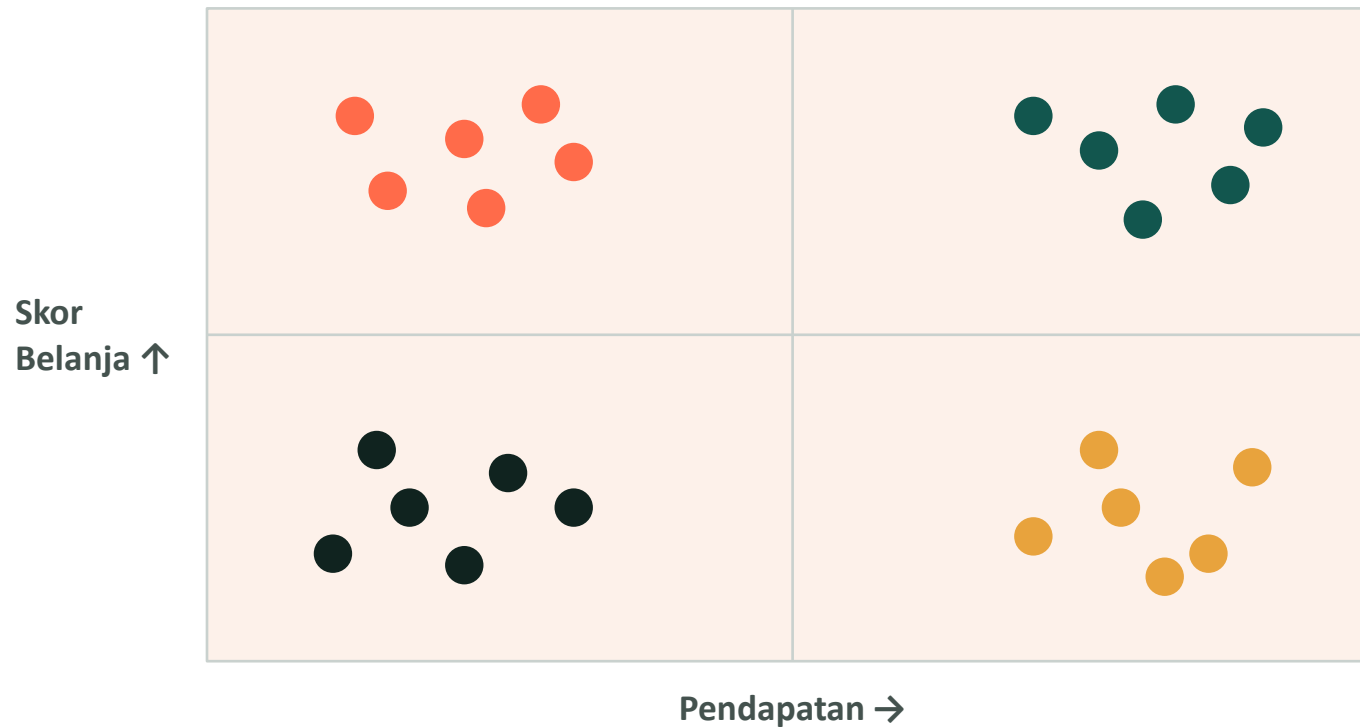
Cara tercepat membaca ciri tiap klaster.

Nomor klaster

Hanya penanda. Klaster 0 tidak lebih baik dari 3.

Peta Segmen Pelanggan

Setiap titik adalah satu pelanggan, diwarnai menurut klasternya.



LEGENDA KLASSTER

- Pendapatan tinggi, belanja tinggi
- Pendapatan rendah, belanja tinggi
- Pendapatan tinggi, belanja rendah
- Pendapatan rendah, belanja rendah

Data ilustrasi; plot nyata dibuat dengan plt.scatter().

Dari Klaster Menjadi Segmen yang Bisa Ditindaklanjuti

Model hanya memberi nomor. Manusia yang memberi makna.

Klaster	Pendapatan	Belanja	Nama Segmen	Strategi Pemasaran
A	Tinggi	Tinggi	Pelanggan Prioritas	Program loyalitas dan akses produk eksklusif
B	Rendah	Tinggi	Pembelanja Impulsif	Promo cicilan dan bundling harga hemat
C	Tinggi	Rendah	Potensial Belum Tergarap	Rekomendasi personal untuk mendorong belanja
D	Rendah	Rendah	Pelanggan Pasif	Kampanye ringan berbiaya rendah

Perhatikan: nama segmen dan strategi adalah penafsiran. Dua analis bisa memberi nama berbeda untuk klaster yang sama.

Seberapa Baik Klaster yang Terbentuk?

Tanpa label, kita tidak punya akurasi. Silhouette mengukur seberapa rapat dan terpisah klasternya.

< 0 Salah klaster Banyak titik lebih dekat ke klaster lain.	0 – 0.25 Lemah Klaster tumpang tindih.	0.25 – 0.5 Cukup Struktur ada, tapi belum tegas.	0.5 – 0.75 Baik Klaster jelas terpisah.	> 0.75 Sangat kuat Hampir tanpa tumpang tindih.
--	--	--	---	--

Contoh kita: $k = 4$ menghasilkan silhouette 0.55, masuk kategori Baik. Ingat, skor tinggi belum menjamin segmennya berguna secara bisnis.

Mencoba DBSCAN pada Dataset yang Sama

Tanpa k. Dua parameter menggantikannya: eps dan min_samples.

```
from sklearn.cluster import DBSCAN

db = DBSCAN(eps=0.5, min_samples=5)
df["db_klaster"] = db.fit_predict(X_scaled)
print(df["db_klaster"].value_counts()) # label -1 = noise
```

eps

Jarak maksimum dua titik dianggap bertetangga.

min_samples

Jumlah minimum tetangga agar titik jadi inti klaster.

Label -1

Titik yang tidak masuk klaster mana pun (noise).

Association Rule pada Data Transaksi

Dataset berbeda: tiap baris adalah satu keranjang belanja, bukan satu pelanggan.

```
from mlxtend.frequent_patterns import apriori, association_rules

# basket: tabel True/False, kolom = nama produk, baris = transaksi
frequent = apriori(basket, min_support=0.1, use_colnames=True)
rules = association_rules(frequent, metric="confidence", min_threshold=0.6)
```

support

Seberapa sering kombinasi item muncul di seluruh transaksi.

confidence

Dari transaksi yang memuat A, berapa persen juga memuat B.

lift

Lebih dari 1 berarti A dan B muncul bersama lebih sering dari kebetulan.

Kesalahan Umum Saat Menerapkan Clustering

Hasil yang "terlihat bagus" belum tentu benar.

Lupa scaling

Fitur berskala besar mendominasi jarak dan klaster menjadi bias.

Memilih k asal

Selalu dukung dengan elbow, silhouette, dan logika bisnis.

Menganggap klaster = kebenaran

Klaster adalah hipotesis, bukan fakta. Uji maknanya bersama pakar.

Fitur tidak relevan

Memasukkan semua kolom, termasuk ID, merusak hasil.

Giliran Anda

Kerjakan dengan dataset pelanggan yang dibagikan, atau dataset pilihan sendiri.

01 Muat dan eksplorasi

Tampilkan head, info, describe. Catat satu temuan.

02 Scaling dan pilih k

Jalankan loop $k = 2-6$. Pilih k dan jelaskan alasannya.

03 Latih dan tafsirkan

Beri nama tiap kluster dan usulkan satu strategi.

04 Bandingkan dengan DBSCAN

Berapa kluster dan noise yang muncul? Mana yang lebih masuk akal?

Rangkuman dan Persiapan UTS

YANG SUDAH DIKERJAKAN

- Alur enam langkah pada dataset contoh
- Scaling, elbow, silhouette, K-Means
- Menafsirkan kluster menjadi segmen

BONUS ALGORITMA

- DBSCAN tanpa menentukan k
- Apriori: support, confidence, lift
- Kesalahan umum yang harus dihindari

PERSIAPAN PERTEMUAN 08

- Minggu 8 adalah UTS (bobot 40%)
- Ulas materi Pertemuan 1–7
- Latih ulang kode K-Means sendiri