# Data Object & Attribute Types

Kumpulan data terdiri dari objek-objek data. **Objek data** biasanya dijelaskan oleh atribut. Objek data juga bisa berupa disebut sebagai sampel, contoh, contoh, titik data, atau objek. Ketika objek data disimpan dalam database, objek tersebut menjadi tupel data. Dengan kata lain, baris database berhubungan dengan objek data, dan kolom berhubungan dengan atribut. Berikut ini akan mendefinisikan atribut dan menjelaskan/memahami berbagai jenis atribut.

## Apa itu Attribute?

**Atribut** adalah bidang data yang mewakili fitur atau karakteristik suatu objek data. Atribut kata benda, dimensi, fitur, dan variabel sering digunakan secara bergantian dalam literatur. Para profesional data mining dan basis data biasanya menggunakan istilah atribut. Atribut yang mendeskripsikan objek pelanggan dapat mencakup, misalnya, ID pelanggan, nama, dan alamat. Jenis atribut ditentukan oleh kumpulan nilai yang mungkin untuk atribut tersebut (nominal, biner, ordinal, atau numerik). Pada penjelasan berikut ini akan menjelaskan berbagai atribut tersebut.

## Nominal Attribute

**Nominal** berarti "*berkaitan dengan sebuah nama*". Nilai **atribut nominal** merupakan *simbol* atau *nama objek*. **Atribut nominal** disebut juga **kategorikal** karena setiap nilai mewakili beberapa jenis kategori, kode, atau status. Nilai-nilai tersebut tidak memiliki urutan yang berarti. Dalam ilmu komputer, nilai disebut juga **enumerasi**.

## Binary Attribute

**Atribut biner** adalah atribut nominal yang hanya memiliki dua kategori atau status (0 atau 1). Biasanya, 0 berarti atribut tidak ada dan 1 berarti atribut ada. Atribut biner dikatakan **Boolean** jika kedua keadaannya berhubungan dengan benar dan salah.

## Ordinal Attribute

**Atribut ordinal** adalah atribut yang nilai-nilainya mungkin memiliki urutan atau peringkat yang bermakna relatif satu sama lain, tetapi ukuran antara nilai-nilai yang berurutan tidak diketahui.

## Numeric Attribute

**Atribut numerik** bersifat kuantitatif, yaitu kuantitas yang dapat diukur, direpresentasikan dalam bilangan bulat atau nilai *real*. Atribut numerik dapat **berskala interval** atau **berskala rasio**.

* **Atribut Berskala Interval (Interval-Scaled Attributes)**

**Atribut skala interval** diukur pada skala satuan yang berukuran sama. Nilai atribut skala interval diurutkan dan bisa **positif**, **0**, atau **negatif**. Oleh karena itu, atribut ini memungkinkan tidak hanya memberi peringkat pada nilai, namun juga membandingkan dan mengukur perbedaan antar nilai.

* **Atribut Berskala Rasio (Ratio-Scaled Attributes)**

Atribut skala rasio adalah atribut numerik yang diberi titik nol. Artinya, jika suatu pengukuran dalam skala rasio, kita dapat mengatakan bahwa suatu nilai merupakan kelipatan (atau rasio) dari nilai yang lain. Selain itu, nilainya diurutkan dan dapat menghitung selisih, mean, median, dan modus antar nilai.

# Mengukur Kemiripan dan Ketidakmiripan Data

Dalam pengaplikasian data mining, seperti pengelompokan, analisis *outlier*, dan klasifikasi tetangga terdekat (KNN), kita memerlukan cara untuk menilai seberapa mirip atau tidak miripnya objek-objek tersebut jika dibandingkan satu sama lain. Analisis *outlier* menggunakan teknik berbasis pengelompokan untuk mengidentifikasi potensial *outlier* sebagai objek yang sangat berbeda dari yang lain.

Pembahasan ini menyajikan ukuran kemiripan dan ketidakmiripan, yang disebut sebagai ukuran kedekatan. Kemiripan dan ketidakmiripan saling berkaitan. Ukuran kemiripan untuk dua objek, misalkan i dan j, biasanya akan mengembalikan nilai 0 jika objek tersebut tidak mirip. Semakin tinggi nilai kemiripan, semakin besar kemiripan antar objek.

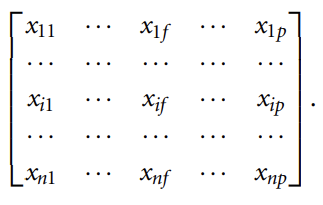
Pembahasan saat ini berhubungan dengan objek yang dideskripsikan oleh lebih dari satu atribut. Kemudian akan membahas bagaimana ketidaksamaan objek dapat dihitung untuk objek yang dideskripsikan dengan atribut nominal (Bagian II.II.), dengan atribut biner (Bagian II.III.), dengan atribut numerik (Bagian II.IV.), dengan atribut ordinal (Bagian II.V.), atau dengan kombinasi dari tipe-tipe atribut itu (Bagian II.VI.). Serta bagian II.VII. memberikan ukuran kemiripan untuk vektor data yang sangat panjang dan jarang, seperti vektor term-frequency yang mewakili dokumen dalam temu kembali informasi (Information Retrieval).

## Data Matrix versus Dissimilarity Matrix

Pada bagian ini, kita akan membahas tentang objek yang dideskripsikan oleh beberapa atribut. Oleh karena itu, kita memerlukan perubahan notasi. Misalkan kita memiliki **n objek** (misalnya, orang, barang, atau mata kuliah) yang dideskripsikan oleh **p** **atribut** (disebut juga pengukuran atau fitur, seperti usia, tinggi, berat, atau jenis kelamin). Objek-objek tersebut adalah x1 = (x11,x12,...,x1p), x2 = (x21,x22,...,x2p), dan seterusnya, dimana xij adalah nilai objek xi dari atribut ke-j. Agar singkatnya, objek xi kita sebut sebagai objek *i* di bawah ini. **Objek** adalah tupel dalam database relasional, juga dikenal sebagai **objek relasional** atau **sampel data** atau **vektor fitur**.

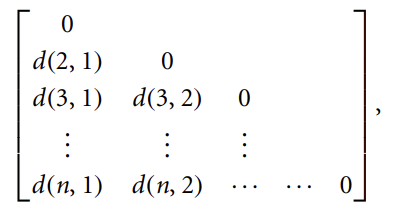
Berikut ini penjelasan dua struktur data, di antaranya:

* **Matriks data/Data Matrix** (atau *struktur objek per atribut*): Struktur ini menyimpan *n* objek data dalam bentuk tabel relasional, atau matriks *n* x *p* (*n* objek × *p* atribut):



Setiap baris berhubungan dengan sebuah objek. Sebagai bagian dari notasi, dapat menggunakan *f* untuk mengindeks atribut *p*.

* **Matriks ketidaksamaan/Dissimilarity Matrix** (atau *struktur objek per objek*): Struktur ini menyimpan koleksi kedekatan yang tersedia untuk semua pasangan *n* objek. Hal ini sering diwakili oleh tabel *n*-by-*n*:



dimana *d*(*i*,*j*) adalah **ketidaksamaan** yang diukur atau “perbedaan” antara objek i dan j. Secara umum, d(*i*, *j*) adalah angka non-negatif yang mendekati 0 ketika objek *i* dan *j* sangat mirip atau “mendekati” satu sama lain, dan menjadi lebih besar jika semakin berbeda. Perhatikan bahwa *d*(*i*, *i*) = 0; artinya, perbedaan antara sebuah objek dengan dirinya sendiri adalah 0. Lebih jauh lagi, *d*(*i*, *j*) = *d*(*j*, *i*). (Untuk memudahkan pembacaan, kami tidak menampilkan entri *d*(*j*,*i*); matriksnya simetris). Ukuran ketidaksamaan akan dibahas pada bagian ini.

Banyak algoritma pengelompokan/*clustering* dan algoritma tetangga terdekat/*nearest-neighbor* yang beroperasi pada matriks ketidaksamaan. Data dalam bentuk matriks data dapat ditransformasikan ke dalam matriks ketidaksamaan sebelum menerapkan algoritma tersebut.

## Proximity Measures for Nominal Attributes

Atribut nominal dapat memiliki beberapa keadaan, misalnya warna peta yang mungkin memiliki lima keadaan: merah, kuning, hijau, pink, dan biru. Jumlah keadaan untuk atribut nominal ini adalah *M*. Keadaan-keadaan tersebut bisa dinyatakan dengan huruf, simbol, atau bilangan bulat (1, 2, ..., Persamaan *M*) yang hanya digunakan untuk penanganan data dan tidak menunjukkan urutan tertentu.

### **Menghitung Dissimilarity**

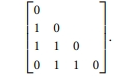
Dissimilarity (ketidakserupaan) antara dua objek *i* dan *j* dihitung dengan formula: *d*(*i*,*j*)=*pp*−*m* di mana:

* + *m* adalah jumlah atribut yang sesuai antara *i* dan *j*,
  + *p* adalah jumlah total atribut.

Jika dua objek memiliki nilai atribut yang sama, maka dissimilarity adalah 0. Jika berbeda, maka dissimilarity adalah 1. Berat tambahan bisa diterapkan untuk meningkatkan efek kesesuaian dalam atribut tertentu yang memiliki lebih banyak keadaan.

### **Contoh Penghitungan Dissimilarity**

Misalkan kita memiliki data dengan satu atribut nominal (test-1) dan empat objek. Matriks dissimilarity untuk objek-objek ini akan terlihat seperti ini:



Dari matriks ini, terlihat bahwa semua objek berbeda kecuali objek 1 dan 4.

### **Menghitung Similarity**

Similaritas (keserupaan) antara dua objek dapat dihitung dengan formula: sim(*i*,*j*)=1−*d*(*i*,*j*)=*pm*

### **Alternatif Pengkodean**

Atribut nominal juga dapat dikodekan menggunakan atribut biner asimetris. Setiap keadaan dari atribut nominal dikonversi menjadi atribut biner baru. Misalnya, untuk atribut warna peta dengan lima keadaan, akan ada lima atribut biner yang masing-masing diatur ke 1 atau 0 sesuai dengan keadaan warna yang dimiliki oleh objek tersebut. Metode pengukuran proksimitas untuk bentuk pengkodean ini dapat dihitung menggunakan metode yang telah dibahas.

## Proximity Measures for Binary Attributes

Atribut biner hanya memiliki dua keadaan: 0 (tidak ada) dan 1 (ada). Pengukuran ketidaksamaan untuk atribut biner dilakukan menggunakan metode khusus karena atribut ini tidak bisa diperlakukan seperti atribut numerik.

**Tabel Kontingensi untuk Atribut Biner**

* **q**: Jumlah atribut yang bernilai 1 untuk kedua objek 𝑖*i* dan 𝑗*j*
* **r**: Jumlah atribut yang bernilai 1 untuk objek 𝑖*i* tapi 0 untuk objek 𝑗*j*
* **s**: Jumlah atribut yang bernilai 0 untuk objek 𝑖*i* tapi 1 untuk objek 𝑗*j*
* **t**: Jumlah atribut yang bernilai 0 untuk kedua objek 𝑖*i* dan 𝑗*j*
* **p**: Jumlah total atribut (p = q + r + s + t)

**Ketidaksamaan untuk Atribut Biner Simetris**

Setiap keadaan dianggap sama penting: 𝑑(𝑖,𝑗)=𝑟+𝑠𝑞+𝑟+𝑠+𝑡*d*(*i*,*j*)=*q*+*r*+*s*+*tr*+*s*

**Ketidaksamaan untuk Atribut Biner Asimetris**

Keadaan 1 (positif) dianggap lebih signifikan daripada keadaan 0 (negatif): 𝑑(𝑖,𝑗)=𝑟+𝑠𝑞+𝑟+𝑠*d*(*i*,*j*)=*q*+*r*+*sr*+*s*

**Kemiripan untuk Atribut Biner Asimetris**

Kemiripan diukur menggunakan koefisien Jaccard: 𝑠𝑖𝑚(𝑖,𝑗)=𝑞𝑞+𝑟+𝑠=1−𝑑(𝑖,𝑗)*sim*(*i*,*j*)=*q*+*r*+*sq* =1−*d*(*i*,*j*)

**Contoh Kasus**

Tabel rekam medis pasien menunjukkan bagaimana ketidaksamaan dihitung antara pasien berdasarkan atribut biner asimetris. Jarak antara setiap pasangan pasien dihitung, menunjukkan bahwa pasien dengan ketidaksamaan tertinggi kemungkinan besar tidak memiliki penyakit yang sama.

Pengukuran ini membantu mengidentifikasi kemiripan atau ketidaksamaan antara objek yang dijelaskan oleh atribut biner, baik simetris maupun asimetris, dengan menggunakan metode seperti koefisien Jaccard untuk kemiripan.

## Dissimilarity of Numeric Data: Minkowski Distance

Dalam bagian ini, dijelaskan ukuran jarak yang umum digunakan untuk menghitung ketidakserupaan objek yang dijelaskan oleh atribut numerik. Ukuran-ukuran ini termasuk jarak Euclidean, Manhattan, dan Minkowski.

Data sering dinormalisasi sebelum menghitung jarak, yang melibatkan mengubah data agar berada dalam rentang yang lebih kecil atau seragam, seperti [-1,1] atau [0,0,1]. Ini bertujuan untuk memberikan bobot yang sama pada semua atribut, meskipun tidak selalu diperlukan dalam setiap aplikasi.

Jarak Euclidean adalah jarak garis lurus antara dua titik. Misalkan 𝑖=(𝑥𝑖1,𝑥𝑖2,...,𝑥𝑖𝑝)i=(xi1 ,xi2 ,...,xip ) dan 𝑗=(𝑥𝑗1,𝑥𝑗2,...,𝑥𝑗𝑝)j=(xj1 ,xj2 ,...,xjp ) adalah dua objek yang dijelaskan oleh 𝑝p atribut numerik. Jarak Euclidean antara objek 𝑖i dan 𝑗j didefinisikan sebagai:



Jarak Manhattan (atau jarak blok kota) adalah total jarak dalam blok antara dua titik di kota. Didefinisikan sebagai:



Kedua jarak Euclidean dan Manhattan memenuhi sifat matematis berikut: non-negatif, identitas yang tidak dapat dibedakan, simetri, dan inequality segitiga. Ukuran yang memenuhi kondisi ini dikenal sebagai metrik.

## Proximity Measures for Ordinal Attributes

Atribut ordinal memiliki urutan atau peringkat yang bermakna, tetapi jarak antara nilai-nilai berurutan tidak diketahui. Contoh atribut ordinal termasuk urutan ukuran seperti kecil, sedang, besar. Atribut ordinal dapat diperoleh dari diskretisasi atribut numerik dengan membagi rentang nilai menjadi kategori terbatas dan terurut.

**Langkah-langkah Penanganan Atribut Ordinal**

1. **Penggantian Nilai dengan Peringkat**:
   * Setiap nilai atribut ordinal diganti dengan peringkatnya yang sesuai.
   * Misalnya, untuk atribut dengan tiga keadaan (adil, baik, sangat baik), nilai-nilai tersebut diberi peringkat 1, 2, dan 3.
2. **Normalisasi Data**:
   * Rentang peringkat dipetakan ke [0.0, 1.0] untuk memberikan bobot yang sama pada setiap atribut ordinal.
   * Rumus normalisasi: 𝑧𝑖𝑓=𝑟𝑖𝑓−1𝑀𝑓−1*zif* =*Mf* −1*rif* −1
3. **Perhitungan Ketidaksamaan**:
   * Ketidaksamaan dihitung menggunakan ukuran jarak yang biasa digunakan untuk atribut numerik, seperti jarak Euclidean, dengan menggunakan nilai normalisasi 𝑧𝑖𝑓*zif* .

**Contoh**

Misalnya, untuk atribut ordinal "tes-2" dengan tiga keadaan (adil, baik, sangat baik), nilai-nilai diberi peringkat dan dinormalisasi menjadi 0.0, 0.5, dan 1.0. Ketidaksamaan antara objek dihitung menggunakan jarak Euclidean, menghasilkan matriks ketidaksamaan.

**Kesimpulan**

Ketidaksamaan untuk atribut ordinal dihitung dengan mengganti nilai dengan peringkat, melakukan normalisasi, dan menggunakan ukuran jarak numerik. Nilai kemiripan dapat diturunkan dari ketidaksamaan dengan rumus 𝑠𝑖𝑚(𝑖,𝑗)=1−𝑑(𝑖,𝑗)*sim*(*i*,*j*)=1−*d*(*i*,*j*).

## Dissimilarity for Attributes of Mixed Types

Bagian ini membahas cara menghitung ketidakserupaan antara objek yang dijelaskan oleh atribut dengan tipe yang berbeda, termasuk nominal, biner simetris, biner asimetris, numerik, atau ordinal. Dalam banyak database nyata, objek sering dijelaskan oleh campuran tipe atribut ini.

Salah satu pendekatan adalah mengelompokkan setiap tipe atribut secara terpisah dan melakukan analisis data mining terpisah (misalnya, clustering) untuk setiap tipe. Namun, dalam aplikasi nyata, hasil analisis terpisah ini jarang kompatibel. Pendekatan yang lebih disukai adalah memproses semua tipe atribut bersama-sama dalam satu analisis. Teknik ini menggabungkan atribut yang berbeda ke dalam satu matriks dissimilarity, dengan menyamakan skala atribut menjadi interval [0.0, 1.0].

Misalkan kumpulan data mengandung 𝑝p atribut dengan tipe campuran. Ketidakserupaan 𝑑(𝑖,𝑗)d(i,j) antara objek 𝑖i dan 𝑗j didefinisikan sebagai berikut:



## Cosine Similarity

Dokumen dapat diwakili oleh ribuan atribut, masing-masing mencatat frekuensi kata atau frasa tertentu dalam dokumen tersebut. Setiap dokumen diwakili oleh apa yang disebut sebagai vektor frekuensi istilah. Misalnya, dalam Tabel 2.5, kita melihat bahwa Dokumen1 mengandung lima instance dari kata "team", sedangkan kata "hockey" muncul tiga kali. Kata "coach" tidak ada dalam seluruh dokumen, yang ditunjukkan dengan nilai hitungan 0. Data semacam ini bisa sangat asimetris.

Vektor frekuensi istilah biasanya sangat panjang dan jarang (memiliki banyak nilai 0). Aplikasi yang menggunakan struktur seperti ini termasuk pengambilan informasi, pengelompokan dokumen teks, taksonomi biologis, dan pemetaan fitur gen. Ukuran jarak tradisional yang telah kita pelajari tidak bekerja dengan baik untuk data numerik yang jarang seperti ini. Misalnya, dua vektor frekuensi istilah mungkin memiliki banyak nilai 0 yang sama, menunjukkan bahwa dokumen yang bersangkutan tidak berbagi banyak kata, tetapi ini tidak membuat mereka serupa. Kita memerlukan ukuran yang fokus pada kata-kata yang dimiliki bersama oleh dua dokumen dan frekuensi kemunculan kata-kata tersebut. Dengan kata lain, kita memerlukan ukuran untuk data numerik yang mengabaikan kecocokan nol.

Cosine similarity adalah ukuran kesamaan yang dapat digunakan untuk membandingkan dokumen atau memberikan peringkat dokumen terhadap vektor kata kunci tertentu. Misalkan 𝑥x dan 𝑦y adalah dua vektor yang akan dibandingkan. Menggunakan ukuran cosine sebagai fungsi kesamaan, kita memiliki:



Secara konseptual, ini adalah panjang dari vektor tersebut. Demikian pula, ∣∣𝑦∣∣∣∣y∣∣ adalah norm Euclidean dari vektor 𝑦y. Ukuran ini menghitung cosinus dari sudut antara vektor 𝑥x dan 𝑦y. Nilai cosinus 0 berarti kedua vektor berada pada sudut 90 derajat satu sama lain (ortogonal) dan tidak memiliki kecocokan. Semakin dekat nilai cosinus ke 1, semakin kecil sudutnya dan semakin besar kecocokan antara vektor. Perlu dicatat bahwa karena ukuran cosine similarity tidak mematuhi semua properti dari ukuran metrik yang didefinisikan dalam bagian 2.4.4, ini disebut sebagai ukuran nonmetrik.