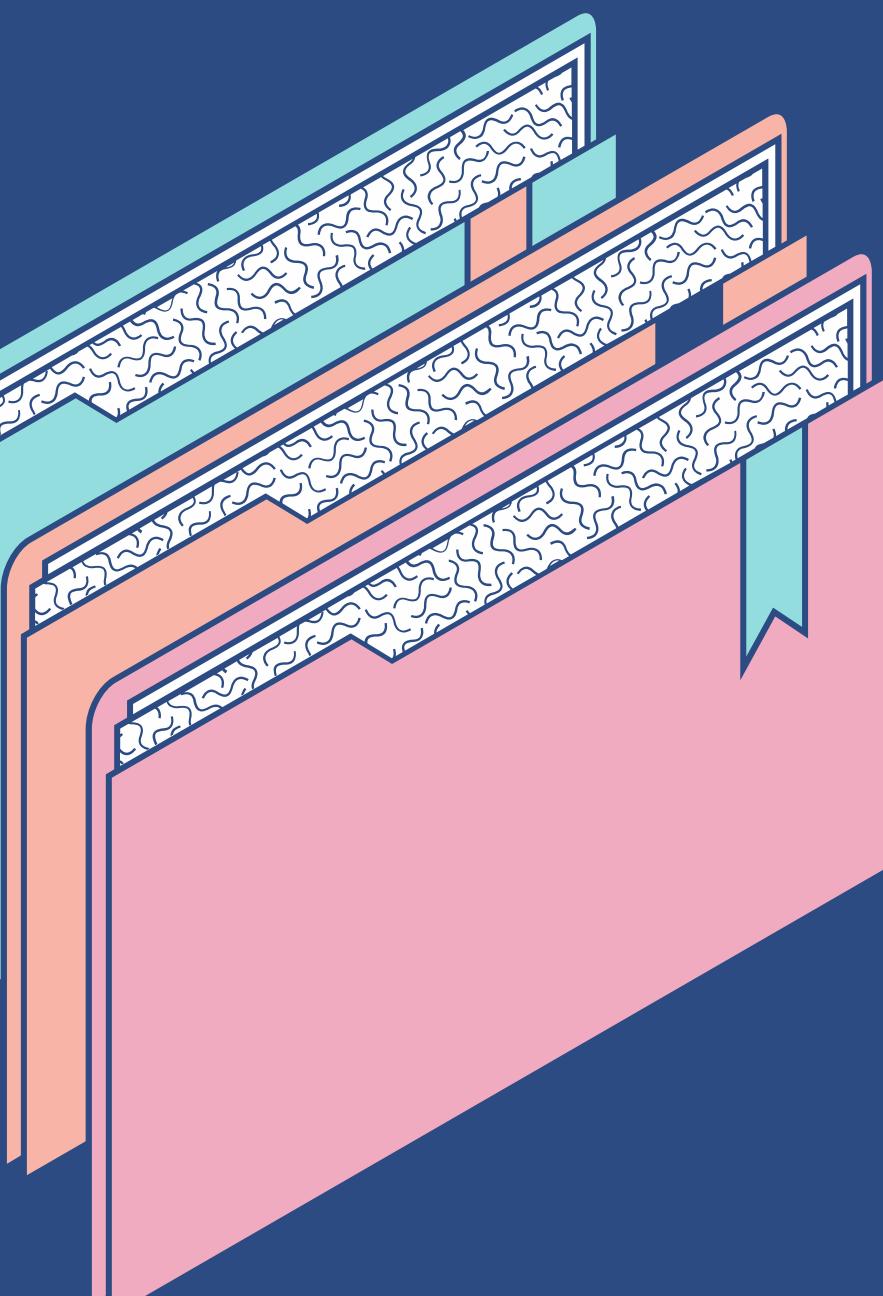


# Konsep Model Data

**Kelompok 5**

- **Hans Valen Putra Wibowo**
- **Nazilma Azzahra**
- **Amar Farhan Riyadi**





# Model Data Hierarkis

Model data hierarkis merupakan model data paling awal yang disusun dalam bentuk pohon (tree).

Setiap induk (parent) dapat memiliki beberapa anak (child), tetapi setiap anak hanya memiliki satu induk.

Struktur ini cocok untuk hubungan satu ke banyak (one-to-many).

Contoh:

Dalam sistem penggajian, tabel Karyawan sebagai induk dan tabel Komputer sebagai anak.

Setiap komputer digunakan oleh satu karyawan tertentu.

# Model Data Jaringan (Network Data Model)

Model data jaringan adalah model basis data yang memungkinkan satu record anak memiliki lebih dari satu induk, sehingga hubungan antar data menjadi lebih fleksibel dan dapat membentuk struktur jaring (network).

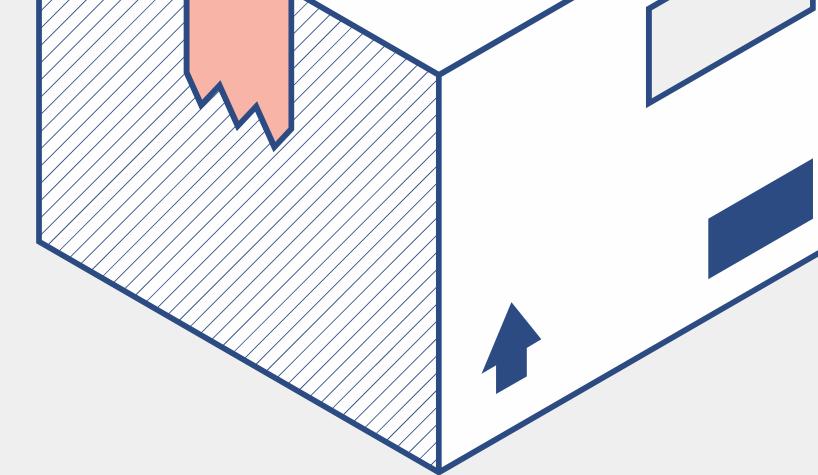
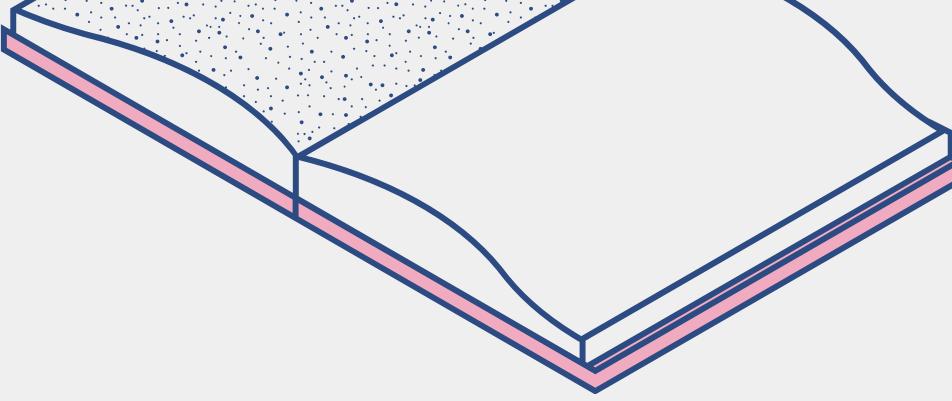
Istilah penting:

- Owner (induk) → record utama.
- Member (anggota) → record yang terkait dengan induk.

Contoh:

Dalam sistem universitas, mahasiswa dapat mengambil banyak mata kuliah, dan setiap mata kuliah dapat diambil oleh banyak mahasiswa. Hubungan semacam ini disebut many-to-many, yang dapat ditangani dengan model jaringan.





## Kelebihan Model Jaringan

- Mendukung hubungan banyak ke banyak (many-to-many).
- Lebih fleksibel daripada model hierarkis.
- Akses data lebih cepat karena menggunakan pointer antar record.

## Kekurangan Model Jaringan

- Struktur lebih rumit, sulit dikelola dan dipelihara.
- Masih memiliki keterbatasan fleksibilitas.
- Membutuhkan pemahaman tinggi dari programmer untuk mengimplementasikan dan memodifikasinya.

# Model Data Relasional Objek

Kombinasi antara model berorientasi objek dan model relasional.

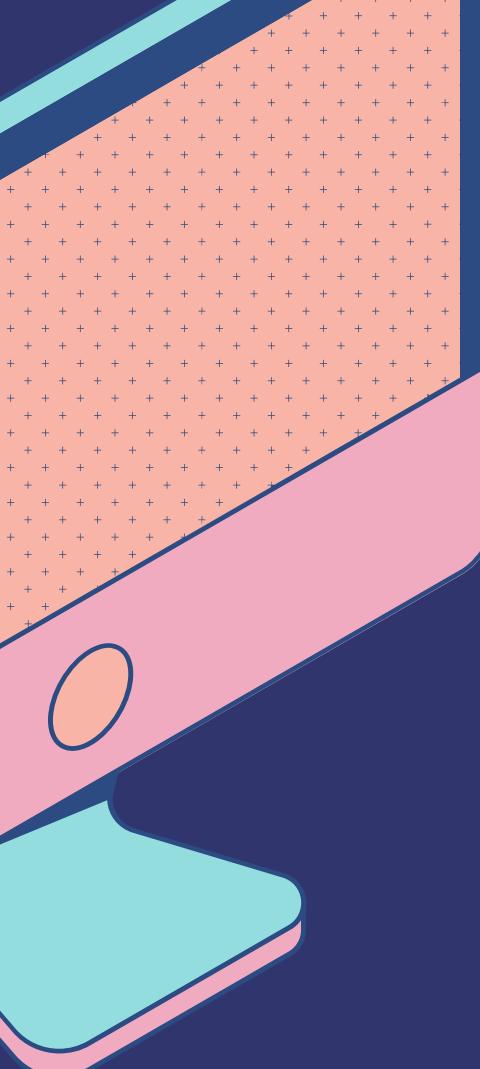
Menyimpan data dalam bentuk tabel, namun juga mendukung objek, kelas, dan pewarisan (inheritance).

Tujuan utama: menghubungkan database relasional dengan bahasa pemrograman seperti C++, C#, dan Java.

Elemen utama dalam model ini:

- Tabel → Menyimpan relasi antar entitas (baris & kolom).
- Tuple → Satu baris data (record).
- Instans Relasi → Kumpulan tuple tanpa duplikasi.
- Skema Relasi → Struktur tabel & atribut.
- Kunci Relasi → Identifikasi unik tiap baris.
- Domain Atribut → Nilai yang diperbolehkan untuk atribut





# Kendala dan Batasan Domain

1. Batasan Domain → Nilai atribut harus sesuai tipe dan rentang (misal usia > 0).
2. Kendala Integritas Referensial → Gunakan foreign key untuk menjaga hubungan antar tabel.
3. Kendala Utama (Primary Key) →
  - Data harus unik, tidak boleh null.
  - Disebut juga kendala entitas.

Bahasa Kueri Database:

- Aljabar Relasional (Prosedural) → Menjelaskan bagaimana mengambil data.

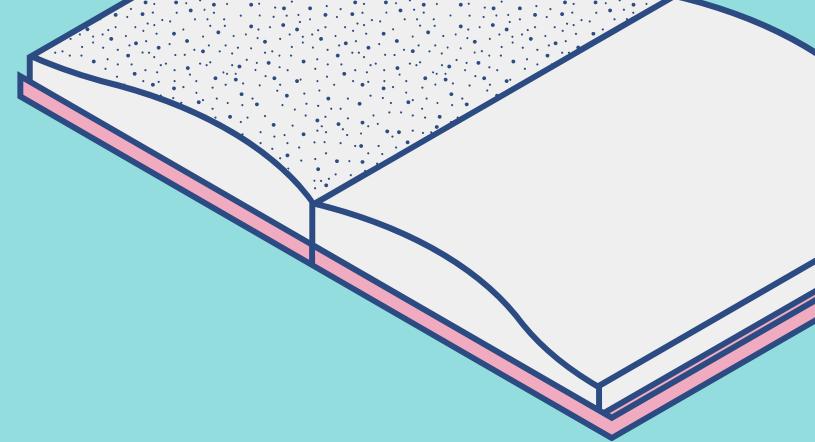
Operasi dasar: Select, Project, Union, Difference, Cartesian Product, Rename.

- Kalkulus Relasional (Non-Prosedural) → Menjelaskan apa yang ingin diambil.

Bentuk:

- a. Tuple Relational Calculus (TRC)
- b. Domain Relational Calculus (DRC)

# Sejarah Model Data Relasional Objek



## Sejarah Singkat:

- Dikembangkan pada tahun 1990-an dari gabungan model relasional dan berorientasi objek.
- Tujuannya: mengatasi kekurangan dari kedua model tersebut.

## Keuntungan:

1. Warisan (Inheritance) → Dapat mewarisi tabel/objek untuk memperluas fungsi.
2. Tipe Data Kompleks → Membentuk tipe data baru dari yang sudah ada.
3. Dapat Diperluas (Extensible) → Mendukung konsep OOP lanjutan seperti pewarisan.

## Kekurangan:

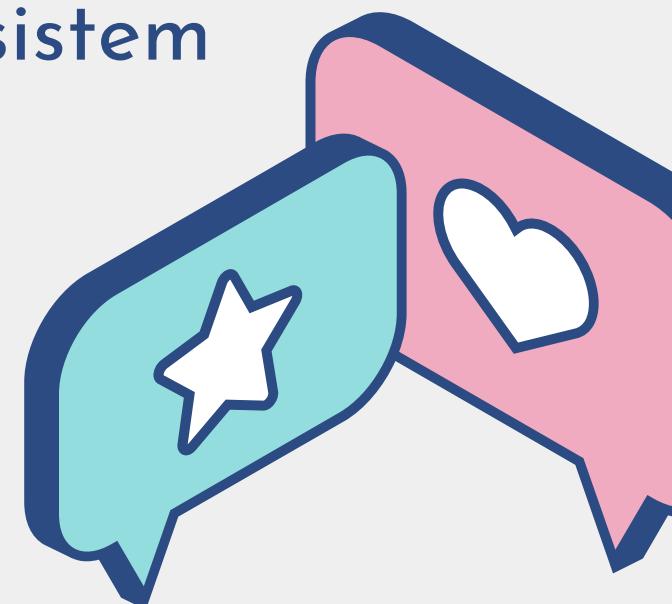
- Struktur bisa rumit dan sulit dikelola karena menggabungkan dua pendekatan yang berbeda.

# Model Relasional dalam DBMS.

Model relasional adalah model data yang merepresentasikan informasi dalam bentuk tabel (relation) yang terdiri dari baris (tuple) dan kolom (attribute). Model ini diperkenalkan oleh Dr. E. F. Codd pada tahun 1970 dan menjadi fondasi utama bagi sebagian besar sistem basis data modern.

Konsep inti model relasional:

- Relation: tabel dua dimensi.
- Tuple: satu baris data.
- Attribute: kolom yang merepresentasikan karakteristik data.
- Domain: himpunan nilai yang valid untuk suatu atribut.
- Primary Key: atribut unik yang mengidentifikasi setiap tuple.
- Foreign Key: atribut yang menunjuk ke primary key tabel lain untuk membentuk relasi.
- Integrity Constraints: aturan untuk menjaga konsistensi dan validitas data (entity integrity, referential integrity, domain integrity).
- Operasi Relasional: menggunakan relational algebra seperti select ( $\sigma$ ), project ( $\pi$ ), join ( $\bowtie$ ), union, intersection, difference.



# Fitur Model Relasional

Model relasional memiliki beberapa fitur utama:

- Representasi tabular: seluruh data disimpan dalam bentuk tabel yang konsisten.
- Bahasa query tingkat tinggi: mendukung SQL untuk manipulasi dan pengambilan data.
- Integrity constraints: menjaga validitas data (primary key, foreign key, domain)



## Keuntungan Model Relasional

- struktur mudah dipahami karena bentuk tabel yang intuitif.
- Akses data fleksibel melalui SQL untuk query kompleks.
- Konsistensi tinggi berkat aturan integritas dan normalisasi.

## Kerugian Model Relasional

- Kurang efisien untuk data non-struktural atau bersarang.
- Join kompleks dapat memperlambat kinerja pada dataset besar.
- Skema kaku, perubahan tabel sering memerlukan migrasi struktur.

# Aturan Codd (Codd's 12 Rules)



- Rule 0 - Foundation: Sistem harus menggunakan model relasional secara eksklusif.
- Rule 1 - Information Rule: Semua informasi direpresentasikan hanya dalam bentuk tabel.
- Rule 2 - Guaranteed Access: Setiap data dapat diakses secara unik menggunakan tabel, primary key, dan nama kolom.
- Rule 3 - Systematic Null Values: Null harus ditangani secara sistematis dan terkontrol.
- Rule 4 - Dynamic Online Catalog: Metadata disimpan dalam tabel dan dapat diakses dengan bahasa query yang sama.
- Rule 5 - Comprehensive Language: Harus ada satu bahasa komprehensif (SQL) untuk definisi data, manipulasi data, dan kontrol transaksi.
- Rule 6 - View Updatability: View yang logis dapat diperbarui bila secara teoritis memungkinkan.
- Rule 7 - High-Level Insert, Update, Delete: Operasi manipulasi data dapat dilakukan terhadap banyak baris sekaligus.
- Rule 8 - Physical Data Independence: Perubahan fisik tidak mempengaruhi aplikasi.
- Rule 9 - Logical Data Independence: Perubahan logis minimal mempengaruhi aplikasi.
- Rule 10 - Integrity Independence: Integrity constraints disimpan dalam katalog dan tidak dalam aplikasi.
- Rule 11 - Distribution Independence: Distribusi data tidak mengubah cara kerja query.
- Rule 12 - Non-Subversion: Tidak ada bypass tingkat rendah yang dapat mengabaikan aturan integritas.

# TERIMA KASIH

