



**Kampus
Merdeka**
INDONESIA JAYA



BASIS DATA MODEL DATA RELASIONAL

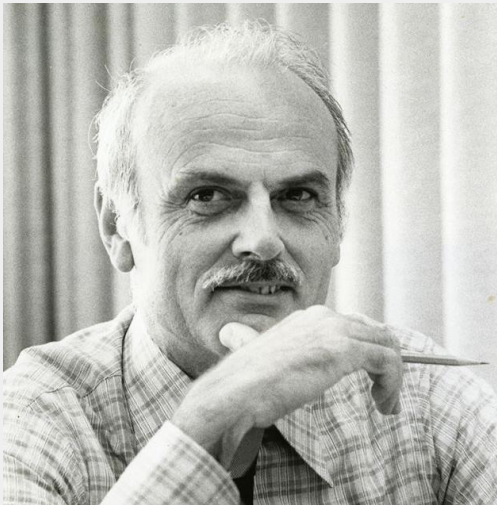
Ir. Abadi Nugroho, M.Kom.
Amelia Yusnita, M.Kom.

Pembelajaran Daring Kolaboratif
STIMIK WICIDA & STITEK BONTANG
TAHUN 2024





Sejarah singkat relation data model

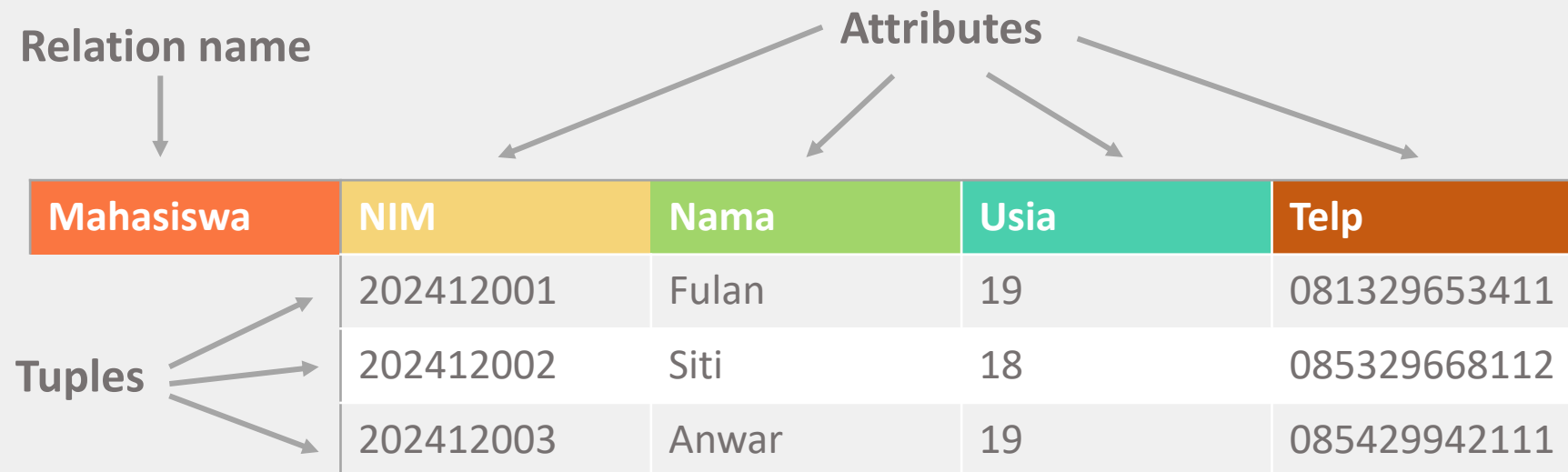


- Pertama kali diperkenalkan oleh Ted Codd (tahun **1970**)
- Menggunakan konsep relasi matematika (**teori himpunan**)
- Pertama kali dikomersilkan oleh Oracle **DBMS** (Database Management System) kemudian SQL/DS (System Structured Query Language/Data System) oleh IBM
- Sekarang populer dengan **RDBMS** (relational database management system) misalnya SQL Server & Access oleh Microsoft, DB2 & Informix oleh IBM dll.
- RDBMS ditulis dengan menggunakan bahasa standar yaitu **SQL query**.



Model Relasional

- Model relasi merepresentasikan data sebagai **kumpulan tabel**.
- Tabel disebut juga **relasi**.
- Setiap baris disebut juga **Tuple**.
- Kolom header disebut sebagai **attribute**.





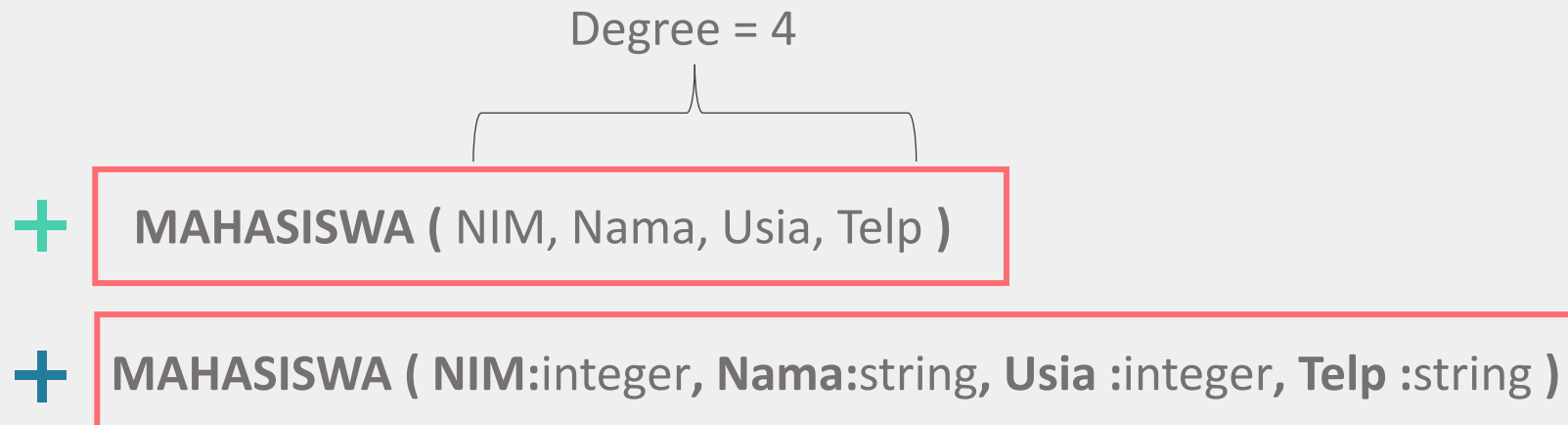
Domain

- Merupakan sebuah nilai yang diperbolehkan untuk sebuah atribut
- Contoh : (mengacu pada gambar tabel di slide sebelumnya)
 - + **NIM** (Nomor Induk Mahasiswa) : berisi karakter angka yang merepresentasikan nomor identitas dari seorang mahasiswa
 - + **Nama** : merupakan sebuah karakter string yang merepresentasikan nama seseorang.
 - + **Usia** : merepresentasikan umur dari mahasiswa yang isinya nanti berupa angka.



Skema Relasi

- Menggambarkan suatu relasi.
- Relasi dilambangkan dengan huruf R dan list atribut A1, A2, A3.... An.
- Contoh : (mengacu pada gambar tabel di slide sebelumnya).



- Degree (or arity) of relation : jumlah atribut (**kolom**) dalam suatu relasi.

Skema Relasi

- **Cardinality** : total jumlah tuple (baris/record) dalam sebuah relasi/tabel

Diagram illustrating the components of a relation schema:

- Relation name**: Points to the table name "Mahasiswa".
- Attributes**: Points to the columns "NIM", "Nama", "Usia", and "Telp".

Mahasiswa	NIM	Nama	Usia	Telp
Cardinality = 3	202412001	Fulan	19	081329653411
	202412002	Siti	18	085329668112
	202412003	Anwar	19	085429942111

- **Relation database schema** : merupakan skema relasi yang ditetapkan dan serangkaian batasan integritas.
- **Relation state (or relation instance)** : sekumpulan tupel pada saat waktu tertentu.



Relasi dengan menggunakan SQL

● Membuat Tabel

```
CREATE TABLE mahasiswa (Nim int,  
                          Nama char(45),  
                          Usia int,  
                          Telp char (12))
```



Relasi dengan menggunakan SQL

- Menyisipkan Tuple/baris

```
INSERT INTO mahasiswa (Nim, Nama, Usia, Telp)  
VALUES ('202412001','Fulan','19','081329653411')
```

- Menghapus Tuple/baris

```
DELETE FROM mahasiswa WHERE Nim= '202412001'
```




Relasi dengan menggunakan SQL

- Mengubah data/nilai

UPDATE mahasiswa

SET Nama='Abadi Nugroho'

WHERE Nim= 202412001



Batasan Integritas

- Merupakan sebuah kondisi yang ditetapkan pada skema dalam database relasional dan akan membatasi data yang dapat disimpan dalam database.
- Batasan integritas dalam basis data bertujuan untuk memastikan **keakuratan** dan **konsistensi** data.

Jenis Batasan integritas

- Batasan kunci (key)
- Batasan kunci asing/tamu (foreign key)
- Batasan Umum



Batasan Integritas

- Batasan Kunci (key)
 - **Primary Key:** Setiap tabel harus memiliki Primary Key yang memastikan setiap baris dalam tabel tersebut unik dan tidak ada duplikasi. Primary Key juga tidak boleh berisi nilai null.
 - **Foreign Key:** Foreign Key digunakan untuk memastikan adanya hubungan yang valid antara tabel yang berbeda. Tujuannya untuk menjaga integritas referensial dengan memastikan bahwa nilai yang direferensikan oleh Foreign Key benar-benar ada dalam tabel lain.



Batasan Integritas

- Batasan Kunci (key)
 - **Unique Key:** Unique Key memastikan bahwa semua nilai dalam kolom tertentu unik, tetapi berbeda dengan Primary Key, Unique Key masih bisa berisi nilai null.
 - **Candidate Key:** Setiap tabel bisa memiliki beberapa Candidate Key, yaitu kolom / kombinasi kolom yang dapat digunakan sebagai Primary Key. Dari beberapa Candidate Key, salah satu dapat dipilih sebagai Primary Key.



Batasan Integritas

- Batasan Kunci (key)
 - **Alternate Key:** Alternate Key adalah Candidate Key yang tidak dipilih sebagai Primary Key.



Batasan Integritas

- Batasan Kunci (key)

Contoh Batasan Kunci (key)

```
CREATE TABLE mahasiswa (Nim int,  
                          Nama char(45),  
                          Usia int,  
                          Telp char (12),  
                          UNIQUE(Nama,Telp),  
                          CONSTRAINT NimMhs PRIMARY KEY(Nim))
```

↓
Nama batasan Primary Key yang ditentukan pengguna.



Batasan Integritas

- Batasan Kunci Asing/tamu (Foreign Key)

- **Referensial Integritas**

Foreign Key harus merujuk pada nilai yang ada dalam tabel induk (tabel yang direferensikan).

Jika nilai dalam tabel induk berubah atau dihapus, maka tindakan yang berkaitan dengan baris-baris dalam tabel anak (tabel yang memiliki Foreign Key) harus sesuai dengan aturan yang ditetapkan (misalnya, cascade, set null, restrict).



Batasan Integritas

- Batasan Kunci Asing/tamu (Foreign Key)
 - **Consistency and Integrity**

Pastikan bahwa hubungan antar tabel tetap valid dan konsisten.

Tujuannya untuk mencegah adanya nilai yang tidak valid atau terisolasi dalam tabel anak.



Batasan Integritas

- Batasan Kunci Asing/tamu (Foreign Key)

- **Cascade Operations**

ON DELETE CASCADE: Jika baris dalam tabel induk dihapus, maka baris terkait dalam tabel anak juga dihapus.

ON UPDATE CASCADE: Jika nilai **Primary Key** dalam tabel induk diperbarui, maka nilai **Foreign Key** dalam tabel anak juga diperbarui.



Batasan Integritas

- Batasan Kunci Asing/tamu (Foreign Key)

- **Restrict and No Action**

RESTRICT: Mencegah penghapusan atau pembaruan nilai yang direferensikan jika ada baris terkait dalam tabel anak.

NO ACTION: Serupa dengan restrict, mencegah tindakan yang melanggar referensial integritas tetapi dieksekusi di akhir pernyataan transaksi.



Batasan Integritas

- Batasan Kunci Asing/tamu (Foreign Key)
 - **Set NULL and Set Default**

SET NULL: Menetapkan nilai null pada Foreign Key di tabel anak jika baris terkait dalam tabel induk dihapus atau diperbarui.

SET DEFAULT: Menetapkan nilai default pada Foreign Key di tabel anak jika baris terkait dalam tabel induk dihapus atau diperbarui.

Batasan Integritas

Batasan Kunci Asing/tamu (Foreign Key)

Contoh :

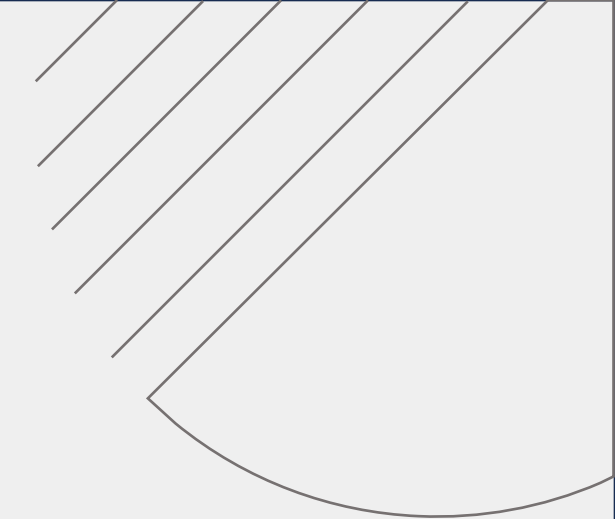
Primary Key

NIM	Nama	Usia	Telp
202412001	Fulan	19	081329653411
202412002	Siti	18	085329668112
202412003	Anwar	19	085429942111

KodeMK	NamaMK	SKS	Semester
TI115	Logika Informatika	3	1
TI321	Sistem Operasi	3	3
RPL746	Sistem Basis Data	4	7

Kunci Asing /tamu/Foreign Key

NIM	KodeMK	Nilai	Semester	TahunAjar
202412001	TI115	80	6	2020/2021
202412002	RPL746	90	6	2020/2021
202412001	RPL746	75	6	2020/2021





Batasan Integritas

- Batasan Umum
 - **Integritas Entitas:** Setiap baris dalam tabel harus dapat diidentifikasi secara unik, biasanya dengan menggunakan Primary Key.
 - **Integritas Referensial:** Menjaga hubungan yang valid antara tabel dengan menggunakan Foreign Key. Misalnya, tidak boleh ada baris dalam tabel anak yang merujuk ke baris dalam tabel induk yang tidak ada / kosong.



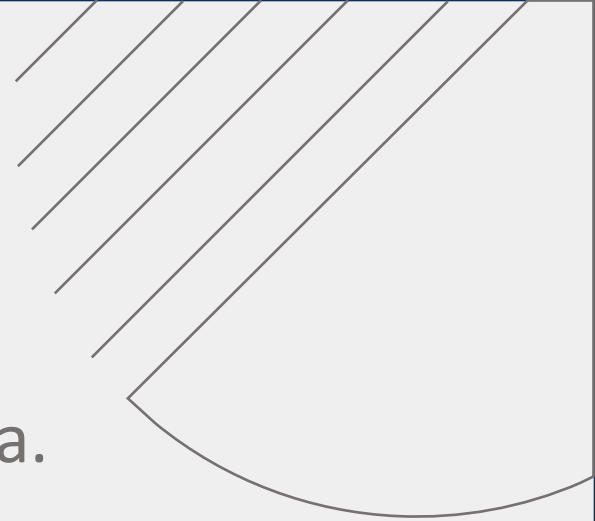
Batasan Integritas

- Batasan Umum
 - **Integritas Domain:** Memastikan bahwa data dalam kolom sesuai dengan jenis data dan batasan yang telah ditetapkan. Contohnya, kolom umur harus berisi angka positif.
 - **Integritas Keunikan:** Kolom atau kombinasi kolom tertentu tidak boleh memiliki nilai yang berulang, diterapkan dengan menggunakan **Unique Key**.



Query Data Relational

- Query merupakan sebuah pertanyaan mengenai data.
- Dalam penulisan Query menggunakan Bahasa khusus.
- SQL merupakan Bahasa query komersial yang paling populer untuk SMBD (Sistem Manajemen Basisdata) relational.





Query Data Relational

Contoh :

```
SELECT *
```

```
FROM AmbilMK
```

```
WHERE KodeMK = 'RPL746';
```

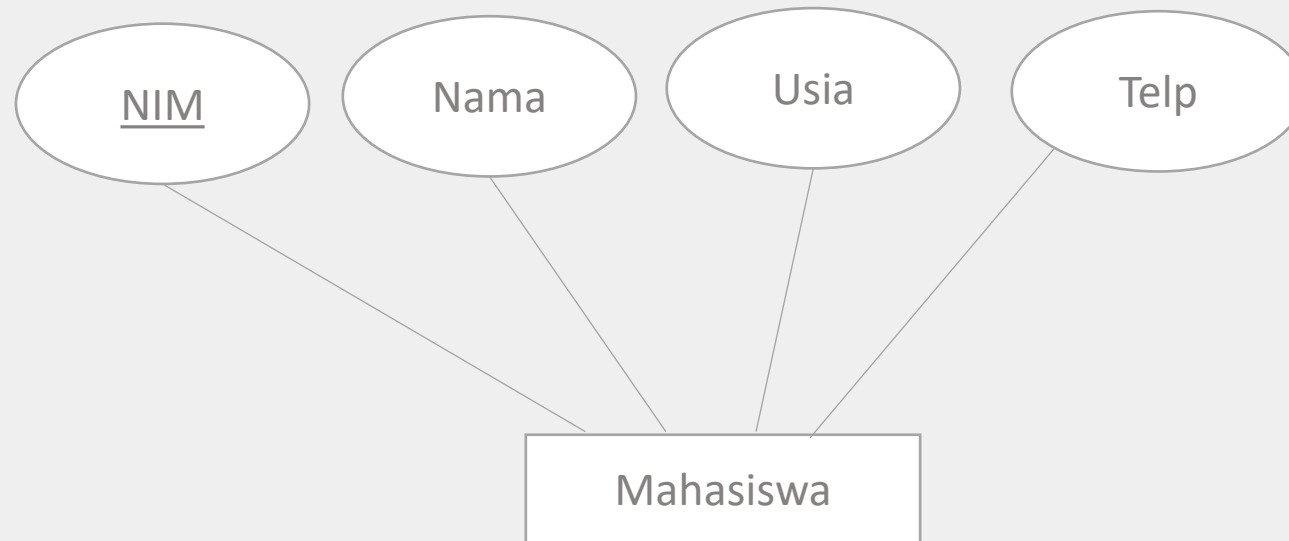
Output

NIM	KodeMK	Nilai	Semester	TahunAjar
202412002	RPL746	90	6	2020/2021
202412001	RPL746	75	6	2020/2021



Desain Database Logis dari ER-Diagram ke Relasional

- Model ER-Diagram sesuai untuk menggambarkan desain database awal.





Desain Database Logis dari ER-Diagram ke Relasional

- SQL (Structured Query Language)

```
CREATE TABLE mahasiswa (Nim int,  
Nama char(45),  
Usia int,  
Telp char (12)  
UNIQUE(Nama,Telp),  
CONSTRAINT NimMhs PRIMARY KEY(Nim))
```

Sample

NIM	Nama	Usia	Telp
202412001	Fulan	19	081329653411
202412002	Siti	18	085329668112
202412003	Anwar	19	085429942111



Kardinalitas (Derajat Relasi)

- Kardinalitas dalam basis data merujuk pada jenis dan jumlah hubungan antara elemen-elemen di dua tabel yang berbeda.
- Konsep ini penting dalam merancang basis data relasional yang efektif dan efisien



Kardinalitas (Derajat Relasi)

● Jenis-jenis Kardinalitas:

Satu-ke-Satu (One-to-One)

- Setiap baris di tabel A berhubungan dengan satu baris di tabel B, dan sebaliknya.
- Contoh: Satu orang memiliki satu paspor.



Kardinalitas (Derajat Relasi)

● Contoh : Satu-ke-Satu (One-to-One)

Satu orang memiliki satu paspor.

```
CREATE TABLE Person (  
    PersonID INT PRIMARY KEY,  
    Name VARCHAR(255)  
);
```

```
CREATE TABLE Passport (  
    PassportID INT PRIMARY KEY,  
    PersonID INT,  
    FOREIGN KEY (PersonID) REFERENCES Person(PersonID)  
);
```



Kardinalitas (Derajat Relasi)

● Jenis-jenis Kardinalitas:

Satu-ke-Banyak (One-to-Many)

- Setiap baris di tabel A dapat berhubungan dengan banyak baris di tabel B, tetapi setiap baris di tabel B hanya berhubungan dengan satu baris di tabel A.
- Contoh: Satu departemen memiliki banyak karyawan.



Kardinalitas (Derajat Relasi)

- Contoh : Satu-ke-Banyak (One-to-Many) Satu departemen memiliki banyak karyawan

```
CREATE TABLE Department (  
    DepartmentID INT PRIMARY KEY,  
    DepartmentName VARCHAR(255)  
);
```

```
CREATE TABLE Employee (  
    EmployeeID INT PRIMARY KEY,  
    DepartmentID INT,  
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)  
);
```



Kardinalitas (Derajat Relasi)

● Jenis-jenis Kardinalitas:

Banyak-ke-Satu (Many-to-One)

- Banyak baris di tabel A dapat berhubungan dengan satu baris di tabel B, tetapi setiap baris di tabel B hanya berhubungan dengan satu baris di tabel A.
- Contoh: Banyak pesanan untuk satu pelanggan.



Kardinalitas (Derajat Relasi)

- Jenis-jenis Kardinalitas:

Banyak-ke-Banyak (Many-to-Many)

- Banyak baris di tabel A dapat berhubungan dengan banyak baris di tabel B, dan sebaliknya.
- Contoh: Banyak siswa mengikuti banyak kursus.



Kardinalitas (Derajat Relasi)

● Contoh : Banyak-ke-Banyak (Many-to-Many)

Banyak siswa mengikuti banyak kursus.

```
CREATE TABLE Student (  
    StudentID INT PRIMARY KEY, StudentName VARCHAR(255)  
);  
  
CREATE TABLE Course (  
    CourseID INT PRIMARY KEY, CourseName VARCHAR(255)  
);  
  
CREATE TABLE Enrollment (  
    StudentID INT,  
    CourseID INT,  
    PRIMARY KEY (StudentID, CourseID),  
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),  
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)  
);
```



**Kampus
Merdeka**
INDONESIA JAYA



TERIMA KASIH

Pembelajaran Daring Kolaboratif
STIMIK WICIDA & STITEK BONTANG
TAHUN 2024