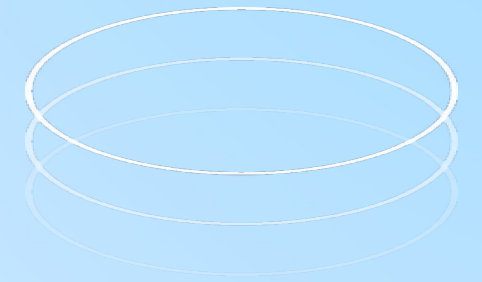


# Model Relasional



BY SATRIA ARDI PERDANA

# OVERVIEW



- Konsep Model Relasional
- Contoh Tabel dan Keterhubungannya
- Istilah Istilah Dalam Model Relational
- Relasional Keys
- Relasional Integrity Rules
- Bahasa Dalam Model Relational
- ERD
- Repeat Data Dictionary
- Subquery
- Q & A
- Challenge / Task

# **Konsep Model Relasional**

1. **Model relasional** : suatu model database yang menggunakan tabel 2 dimensi, yang terdiri dari row/baris/record/tuple dan kolom/field/atribut untuk menggambarkan sebuah database.
2. Model paling sederhana dan banyak digunakan pada saat ini.
3. Karena sederhana seorang programmer / pengembang aplikasi akan lebih mudah dalam melakukan operasi database. Misalnya proses CRUD (Create, Read, Update, Delete) database.

**4. Model relasional :** suatu model database yang menggunakan tabel 2 dimensi, yang terdiri dari row/baris/ record/tuple dan kolom/field/ atribut untuk menggambarkan sebuah database.

5. Diperkenalkan oleh peneliti di IBM Research Laboratory E.F. Codd tahun 1970 -an,  
<https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>

## **Keuntungan model relasional:**

1. Dapat mengakomodasi kebutuhan pengelolaan database yang ada di dunia nyata, misal: pengelolaan data member, data transaksi, data barang dan sebagainya.
2. Model paling sederhana dan banyak digunakan pada saat ini.



## **Lanjutan keuntungan model relasional:**

3. Karena sederhana seorang programmer / pengembang aplikasi akan lebih mudah dalam melakukan operasi basis data CRUD (Create, Read, Update, Delete) pada sistem yang dikembangkannya.
4. Pencarian data pada 1 tabel atau banyak tabel bisa dilakukan dengan cepat.

# Contoh tabel dan relasinya

Member

tbmm_id	tbmm_name	tbmm_address	tbmm_gender	tbmm_phone
1	Ridwan Santoso	Jl. Srigala no 23	Pria	086543823
2	Ani	Jl. Landak no 43	Wanita	08765444

Product

tbmp_id	tbmp_name	tbmp_price	tbmp_stock
1	Air Mineral A	4000	200
2	Permen A	3000	2000
3	Mie Goreng A	5000	100
4	Mie Goreng B	6000	20

Order

tbto_id	tbto_created_date	tbto_tbmm_id	tbto_tbmp_id	tbto_qty
1	202400509	1	1	3
2	202400508	2	1	1



# **Istilah Dalam Model Relasional**

## **Istilah dalam model relasional:**

**1.Relasi**

**2.Atribut**

**3.Tuple**

**4.Domain**

**5.Derajat/Degree**

**6.Cardinality**

**7.Relation Key (Super Key, Kandidat Key, Primary Key, Alternate Key, Foreign Key).**

**8.Relation Integrity Rule (null, Entity Integrity, Referential Integrity )**

**Relasi** → Sebuah tabel yang terdiri dari beberapa column dan beberapa row.

**Atribut** → Kolom pada sebuah relasi.

**Tupel** → Baris pada sebuah relasi.

**Domain** → kumpulan nilai yang valid untuk satu atau lebih atribut

**Derajat/Degree** → jumlah atribut dalam sebuah relasi.

**Cardinality** → jumlah tupel dalam sebuah relasi.

# Relational Key

## 1. Super Key

Satu atau kumpulan atribut yang secara unique mengidentifikasi sebuah tupel / row dalam relasi. Contoh dalam tabel mahasiswa super keynya:

- NIM
- Nama (dengan syarat nama tidak ada yang sama)
- Alamat (dengan syarat tidak alamat yang sama)
- NIM + Nama
- NIM + Alamat
- NIM + Nama + Alamat

## Lanjutan Relational Key

### 2.Candidate Key

Atribut di dalam relasi yang biasanya mempunyai nilai unique. Contoh pada tabel **Mahasiswa** candidatenya adalah : NIM, Nama dan Alamat.

### 3.Primary Key

Candidate key yang dipilih untuk mengidentifikasi tupel secara unique dalam relasi. Contoh jika NIM sebagai primary key maka nilainya pasti unique atau tidak akan pernah sama.

## Lanjutan Relational Key

### 4.Alternate Key

Candidate key yang tidak dipilih sebagai primary key. Pada tabel **Mahasiswa** jika NIM dipilih jadi primary key, maka **Alternate Keynya** adalah Nama dan Alamat.

### 5.Foreign Key

Atribut dengan domain yang sama yang menjadi primary key pada sebuah relasi, tetapi pada relasi lain atributnya hanya sebagai atribut biasa.



# Relational Integrity Rule

## 1. Null

Nilai suatu atribut yang tidak diketahui dan tidak cocok untuk row/tuple tersebut.

## 2. Entity Integrity

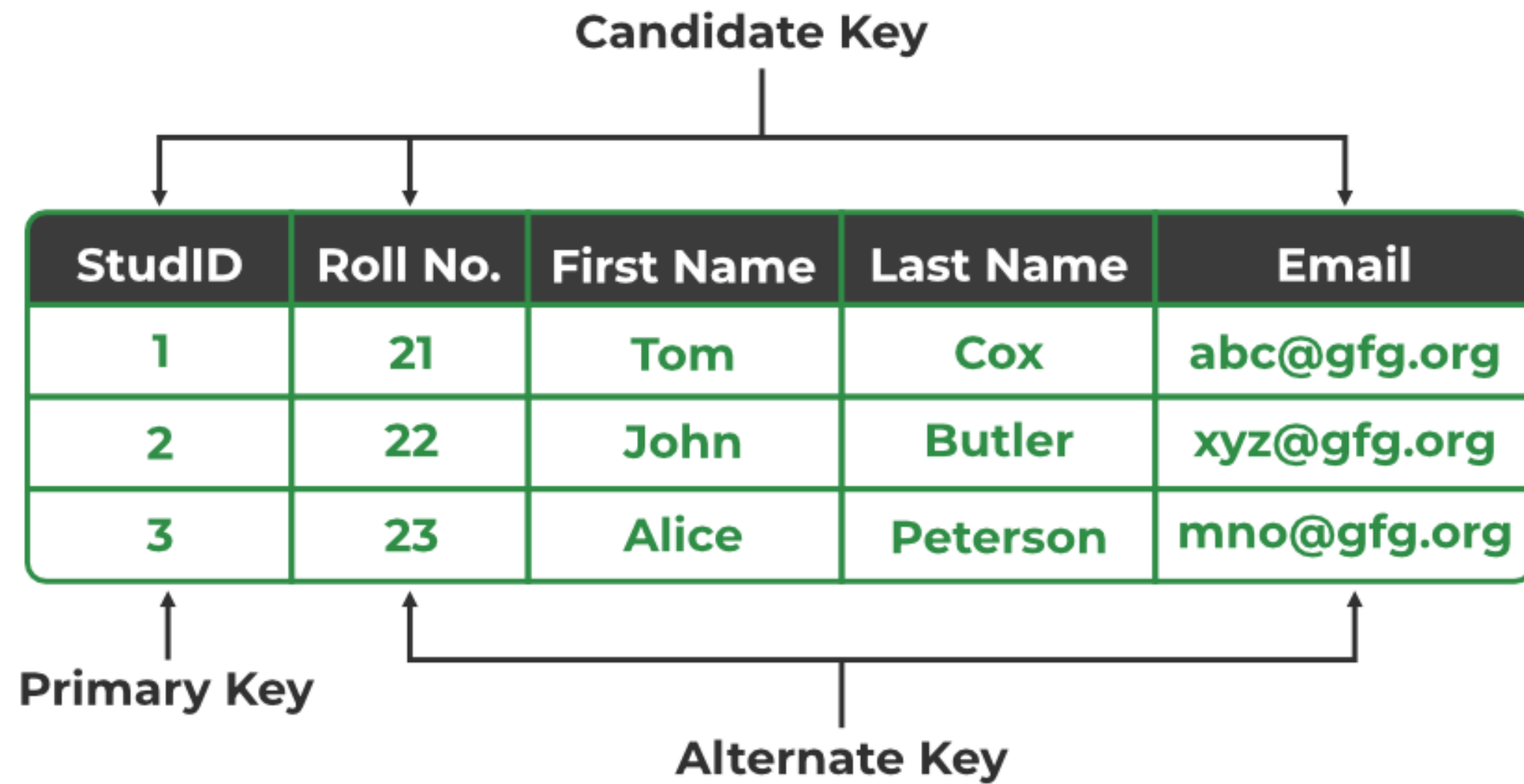
Tidak ada satu komponen primary key bernilai null (PK tidak boleh null karena setiap tabel harus memiliki key yang unique). Contoh: setiap mahasiswa memiliki kode mahasiswa (nim) yang unique sehingga bisa membedakan mahasiswa yang satu dengan yang lainnya.

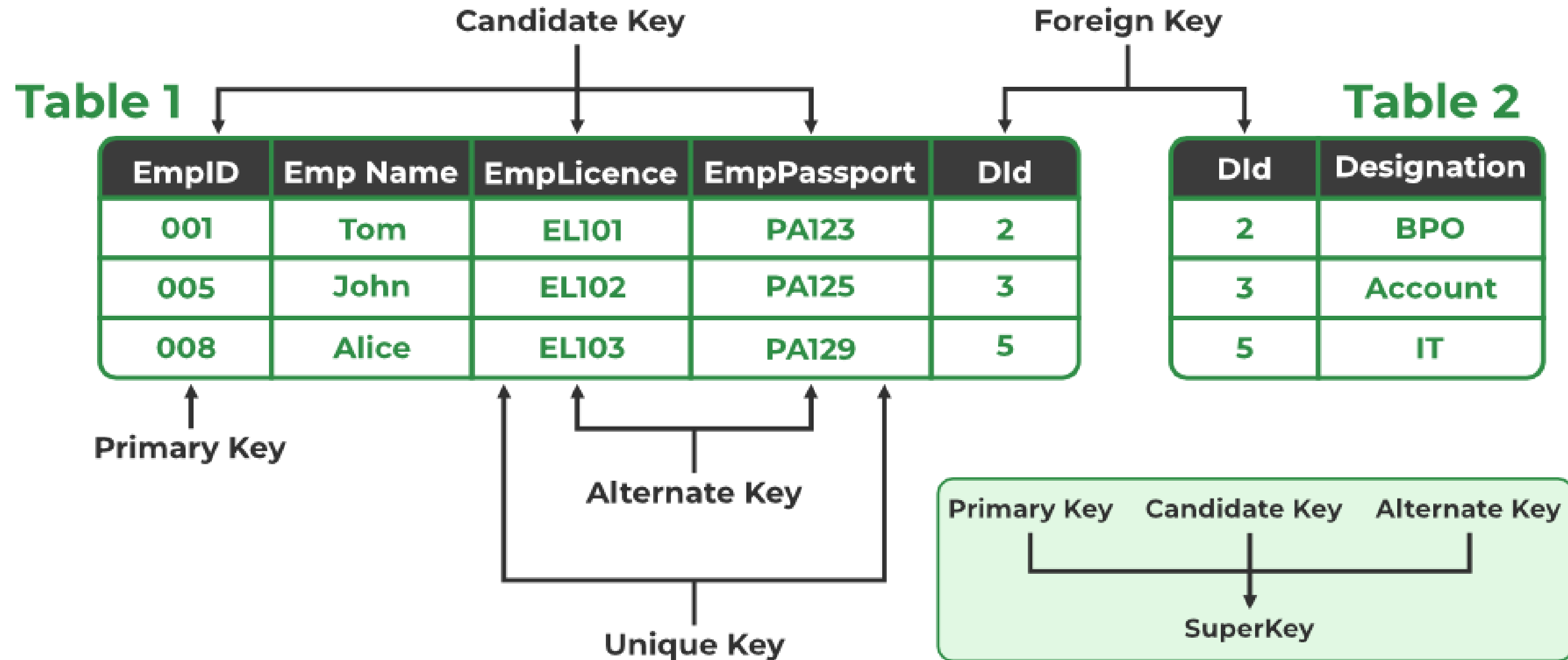
## 3. Referential Integrity

Dapat digambarkan kekonsistenan data antara relasi PK dengan FK. Misal pada tabel **Order** terdapat FK member\_id yang berelasi ke tabel master member. Pada tabel master member disini member\_id sebagai PK. Nilai pada FK tersebut harus ada pada PK

**Note: null beda dengan nol (0).**

## Contoh tabel untuk menjelaskan



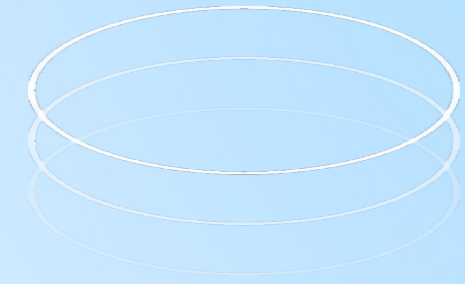


# **Kamus Data / Data**

- **Data dictionary** → penjelasan secara tertulis tentang data yang berada dalam database, contoh kamus data versi lengkap:
- Tabel Master Barang:
- Pk:
- Fk:
- Jumlah Field:
- Contoh dibawah ini tinggal ditambah keterangan.

Column Name	#	Data Type
123 tbmb_id	1	bigint(20)
ABC tbmb_name	2	varchar(100)
123 tbmb_harga	3	int(11)
123 tbmb_stok	4	int(11)

- Kamus





# **Entity Relationship Diagram(ERD)**

## **ERD :**

1. Merupakan diagram yang menggambarkan relasi antar entity atau database secara conceptual dari sudut pandang end user.
2. ERD menekankan pada struktur data dan relasi data.

## **Komponen ERD :**

1.Entity

2.Attribute

3.Relasi

# Entity

Entity merupakan suatu objek atau tempat untuk menyimpan data

Contoh: Member, Mahasiswa, Order, Product

## Attribute

Attribute merupakan karakteristik yang dimiliki suatu entity

Contoh:

1. Member atributnya id\_member, nama\_member, alamat\_member dll.
2. Mahasiswa atributnya nim, nama\_mahasiswa, alamat\_mahasiswa dll
3. Order atributnya id\_order, tanggal\_order id\_customer dll.
4. Product atributnya id\_product, nama\_product, harga\_product, stock dll

## Relasi

Hubungan antara satu entity dengan entity lainnya, yang dihubungkan dengan relasi.

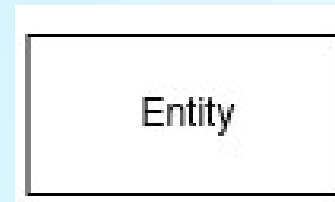
Contoh:

Entity Member mempunyai hubungan dengan Entity Order. Isi hubungannya bisa berupa misalnya member tersebut melakukan order pada tanggal sekian.

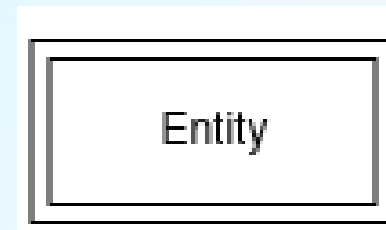


## Symbol pada ERD

### 1.Entity



### 2.Weak entity



### 3.Relationship



### 4.Identifying Relationship



## Lanjutan Symbol pada ERD

### 5.Atribut



### 6.Atribut Primary Key



### 7.Atribut Multivalue



### 8.Atribut Composite



### 9.Atribut Derivatif



## **Best Practice Cara membuat ERD contoh proses transaksi**

1.Tentukan Entity

2.Tentukan Relasi

3.Gambar ERD Sementara

4.Isi kardinalitas

5.Tentukan PK

6.Revisi ERD dengan PK

7.Tentukan Atribut dan masukan ke ERD

8.Periksa hasil, jika perlu dilengkapi kembali ke nomor 1.

# Contoh membuat ERD pada customer membeli product

## 1.Entity

- customer
- product

## 2.Atribut

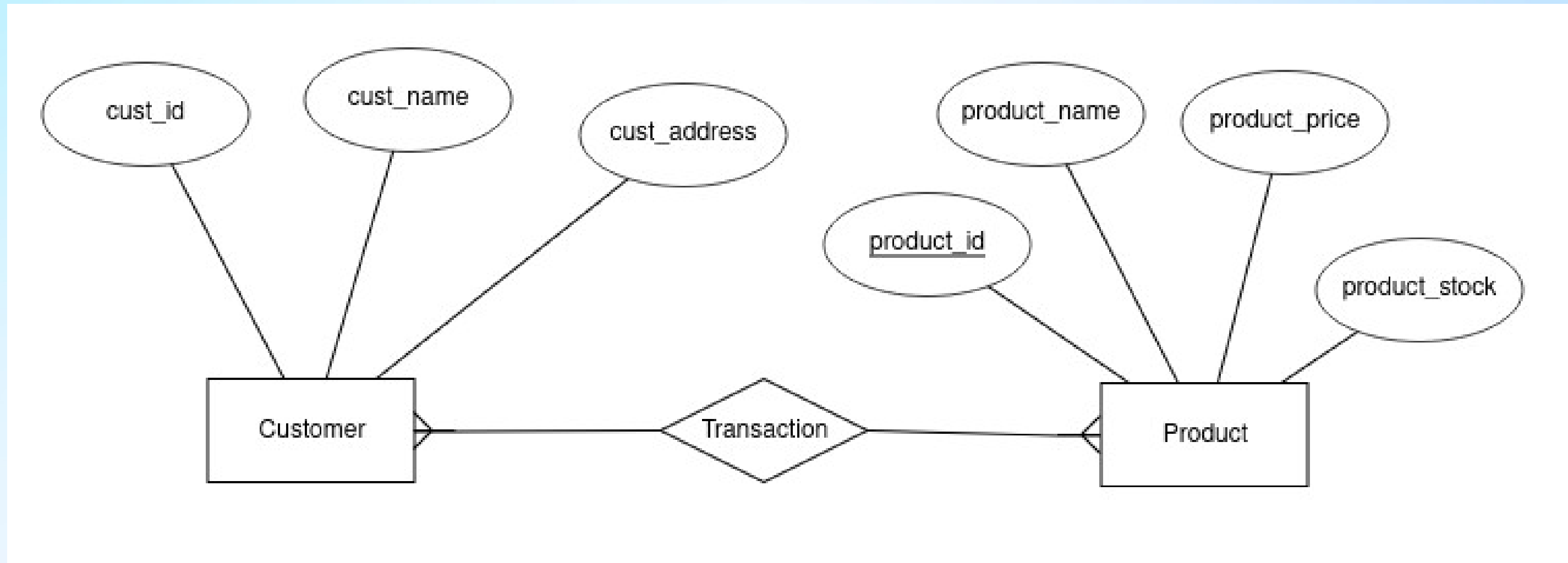
- customer: cust\_id, cust\_name, cust\_address, ...
- product: product\_id, product\_name, product\_price, product\_stock,...

## 3.Relasi

- Banyak customer bisa melakukan transaksi banyak product, tidak hanya 1 customer saja yang boleh transaksi, dan customer bisa transaksi tidak hanya 1 barang saja→ Many to Many.

## Lanjutan contoh membuat ERD pada customer membeli product

### 4.ERD



# **Tambahan Materi - Sub Query**



# Subquery

- SQL sub query adalah adanya query di dalam query utama dan biasanya berada didalam WHERE Clause.
- Sub query dapat digunakan untuk menghasilkan satu set data yang kemudian digunakan oleh query utama
- Sub query dapat berada dalam proses SELECT, INSERT, UPDATE, atau DELETE.
-

# Rule Subquery

- Harus didalam tanda kurung subquerynya

# Tipe Subquery

- Scalar Sub Query
- Single Row Sub Query
- Multiple Row Sub Query
- Correlated Sub Query

# Scalar Sub Query

Scalar subquery akan menghasilkan nilai tunggal.

Umumnya, nilai ini kemudian digunakan pada klausa SELECT, WHERE, ataupun HAVING untuk membandingkan data yang ada pada query utama dengan nilai tunggal yang dihasilkan dari subquery.

```
⊖ SELECT nm_pegawai  
   FROM pegawai  
  WHERE salary < (SELECT AVG(salary) FROM pegawai);
```

# Single Row Sub Query

Single row subquery akan menghasilkan satu nilai output saja.

Sebagai contoh, perintah subquery dibawah ini hanya akan menghasilkan 1 nilai saja.

```
SELECT  
    kd_dosen, nm_dosen  
FROM dosen  
WHERE  
    kd_dosen = (SELECT kd_dosen FROM matakuliah where nama_mk='Praktikum Basis Data');
```

# Multiple Row Sub Query

Multiple row subquery akan menghasilkan lebih dari satu nilai output. Jenis subquery ini biasanya digunakan dalam operator pembandingan seperti IN, ANY, ALL

IN → Sama dengan semua hasilnya (list), misal (101,102,103)

```
SELECT
    kd_dosen,nm_dosen
FROM dosen
WHERE
    kd_dosen in (SELECT kd_dosen FROM matakuliah);
```

# Multiple Row Sub Query

Operator = ANY → IN

Operator < ANY → Kurang dari MAX

Operator > ANY → Lebih dari MIN

```
SELECT  
    *  
FROM matakuliah  
WHERE  
    sks > ANY (SELECT sks FROM matakuliah where semester=3);
```



# Multiple Row Sub Query

Operator < ALL → Kurang dari MIN

Operator > ALL → Lebih dari MAX

```
SELECT  
    *  
FROM matakuliah  
WHERE  
    sks > ALL (SELECT sks FROM matakuliah where semester=3);
```

# Corelated Sub Query

Dalam correlated subquery, inner query akan dilakukan berulang kali berdasarkan kriteria yang sesuai dengan outer query. Ciri utama corelated memiliki sumber data yang sama dengan query utama.

Berikut adalah contoh correlated query.

```
SELECT p.kd_pegawai , p.nm_pegawai  
FROM pegawai p  
WHERE salary < (SELECT AVG(salary) FROM pegawai where kd_pegawai in (101,102));
```

# Penggunaan Subquery Dengan Satement INSERT

Jumlah kolom/field yang di INSERT harus sama dengan yang di SELECT

#Insert Subquery di tabel pegawai2 jumlah kolomnya harus sama dengan tabel pegawai

```
INSERT INTO pegawai2  
(SELECT * FROM pegawai WHERE salary < 6000000);
```

#Insert Subquery di tabel pegawai3 junlahnya harus 2 kolom

```
INSERT INTO pegawai3  
(SELECT nm_pegawai, alamat FROM pegawai WHERE salary < 6000000);
```

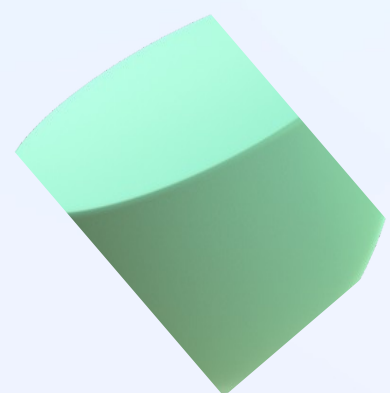
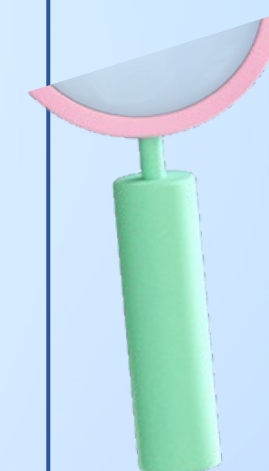
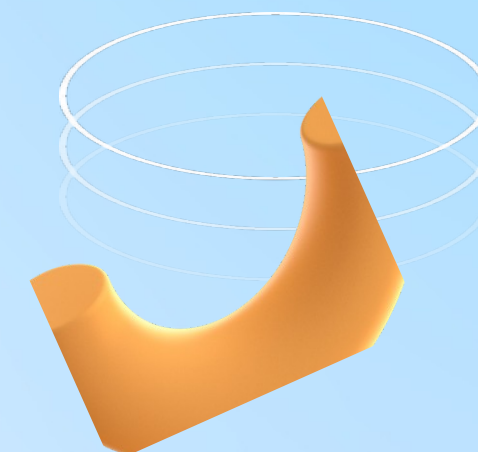
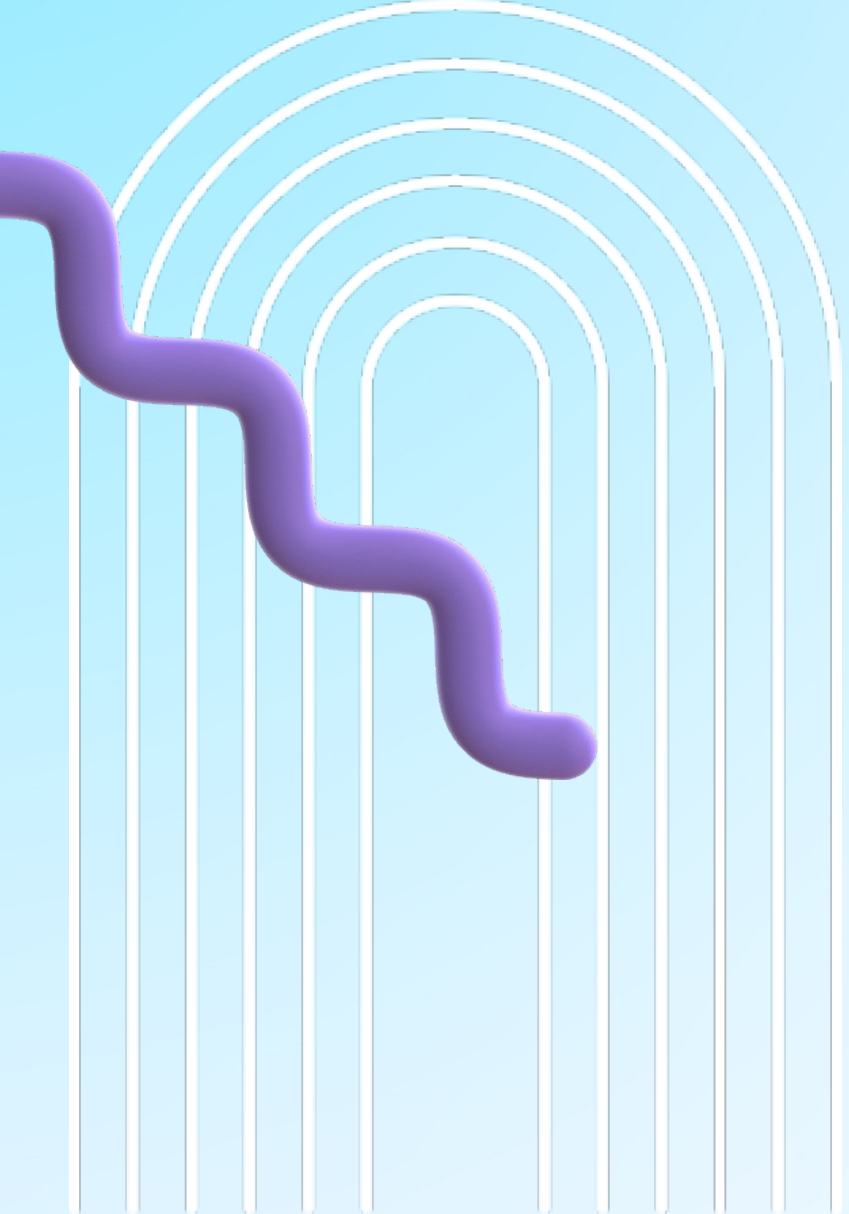
# Penggunaan Subquery Dengan Statement UPDATE

```
#Update Subquery -> update pegawai4 jika salary kurang dari 6000000 berdasarkan table pegawai  
UPDATE pegawai4  
SET salary = 6000000  
WHERE salary IN (SELECT salary FROM pegawai WHERE salary < 6000000 );
```

# Penggunaan Subquery Dengan Klausa DELETE

```
#Delete Subquery
-- pemrogramanbasisdata.pegawai definition
DELETE FROM pegawai5
WHERE salary IN (SELECT salary FROM pegawai WHERE salary > 5000000 );
```

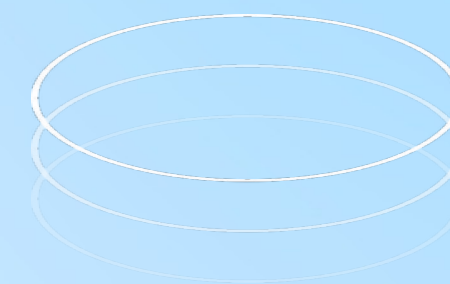
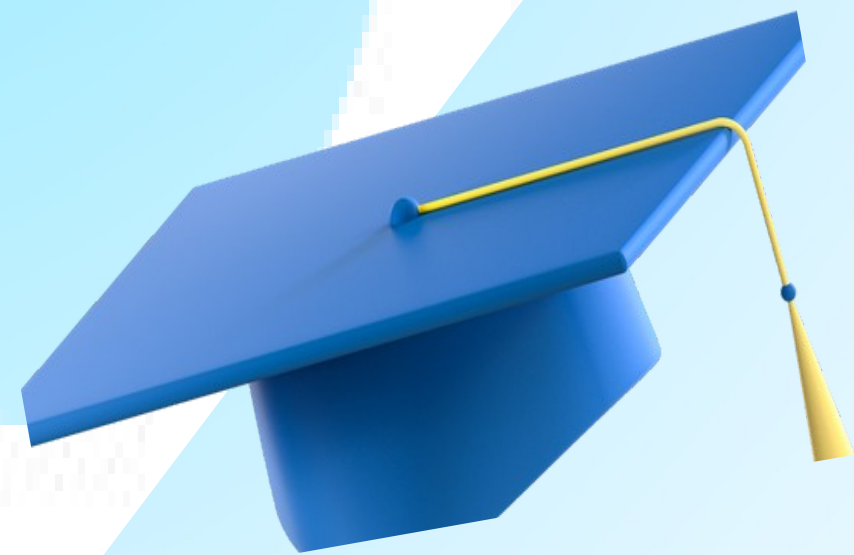
DO YOU HAVE  
ANY QUESTION?



## **Challenge**

**Case Sistem Informasi Akademik misalnya ada Mahasiswa, Dosen, Ruang Kelas, Mata kuliah**

**Buatlah ERD sederhana sesuai langkah langkah yang dicontohkan**



# THANK YOU

FOR YOUR ATTENTION

