



Entity Relationship Diagram

Ahmadi Irmansyah Lubis



Mengapa ER Diagram ?

- ❖ Pada skala tertentu, perancang database dapat langsung memutuskan rancangan basis data yang akan dibuat, relasi, atribut dan constrainnya
- ❖ Namun demikian pada prakteknya, rancangan basis data yang dibutuhkan seringkali kompleks, sulit untuk langsung menentukan rancangan atribut, relasi dan rancangan lainnya.
- ❖ Seringkali tak seorangpun memahami secara menyeluruh kebutuhan data untuk sebuah aplikasi, perancang basis data harus berinteraksi dengan customer untuk memahami kebutuhan aplikasi
- ❖ Dibutuhkan model data yang menerjemahkan kebutuhan menjadi skema konseptual dari basis data

Review : Database Design

Need
Analysis

Data Model
(e.g ER)

Relational
Schema

DBMS
+ Query



Conceptual Design

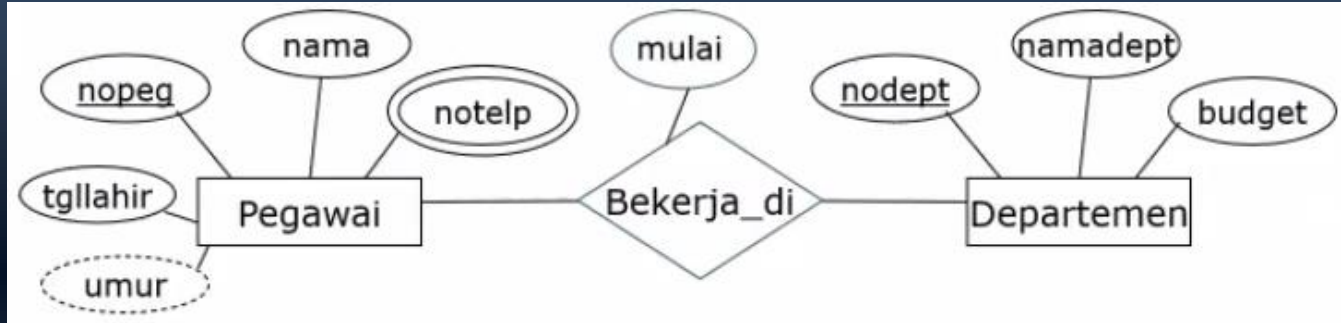
ER Model – teknik pemodelan data yang mengilustrasikan secara visual entitas dalam basis data / sistem informasi dan relasi antara entitas tersebut

ER Model digunakan pada tahap ini:

- ☐ Apa saja **entitas** dan **relationship** antar entitas tersebut dalam sebuah organisasi?
- ☐ Apa saja informasi mengenai entitas dan relationship tersebut yang harus disimpan dalam database?
- ☐ Apa saja *integrity constraints* atau *business rules* yang berlaku?
- ☐ Sebuah 'skema' database (struktur) dalam ER Model dapat digambarkan (ER Diagram)
- ☐ ER Diagram dapat dipetakan menjadi skema relasional






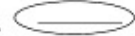

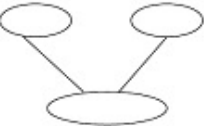



E-R Diagram



- ❖ Persegipanjang melambangkan himpunan entitas
- ❖ Belahketupat melambangkan himpunan relationship
- ❖ Garis menghubungkan atribut ke himpunan entitas dan himpunan entitas ke himpunan relationship
- ❖ Elips melambangkan atribut
 - ❖ Elips dobel melambangkan multi-valued atribut
 - ❖ Elips dengan garis putus-putus melambangkan derived attributte
- ❖ Garis bawah melambangkan atribut primary key

Notasi E-R Diagram

<u>Notasi</u>	<u>Arti</u>
1. 	1. Entity
2. 	2. Weak Entity
3. 	3. Relationship
4. 	4. Identifying Relationship
5. 	5. Atribut
6. 	6. Atribut Primary Key
7. 	7. Atribut Multivalue
8. 	8. Atribut Composite
9. 	9. Atribut Derivatif

Instance Dari Sebuah E-R Diagram

Departemen

Nodept	Namadept	Budget
D100	Farmasi	120.000.000
D101	Kardiologi	540.000.000
D102	HR	20.000.000

Bekerja_di

Namadept	Nopeg	Mulai
D100	123-45-6789	1964
D101	222-33-4444	2005
D102	999-777-5555	1999

- ❖ E-R Diagram mendeskripsikan skema sebuah database
- ❖ Database instance berisi data yang mengikuti struktur tersebut
- ❖ Kita tidak memiliki data dalam conceptual design, tapi membayangkan bahwa data ada membantu kita berpikir mengenai desain

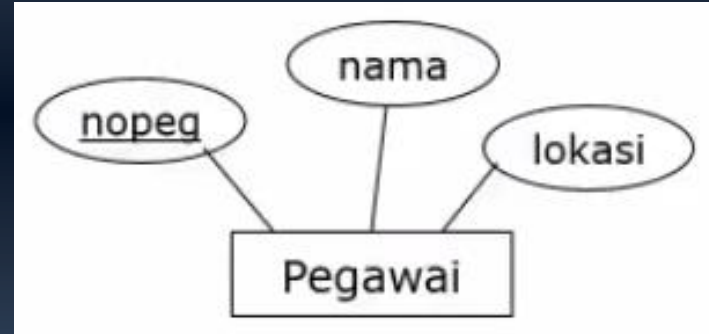
ER Model - Entity

Entitas (atau **entity**): objek dalam dunia nyata yang dapat dibedakan dari objek yang lain

- ❑ Misal: seseorang, sebuah perusahaan, sebuah event
- ❑ Dideskripsikan (dalam database) menggunakan kumpulan **atribut**

Himpunan entitas (atau **entity set**): himpunan entitas-entitas yang serupa yang mempunyai properti yang sama. Misal semua pegawai

- ❑ Semua entitas dalam sebuah himpunan entitas mempunyai atribut yang sama
- ❑ Misal himpunan semua orang, perusahaan, event
- ❑ Setiap entity set memiliki **key**



ER Model - Atribut

Himpunan entitas memiliki:

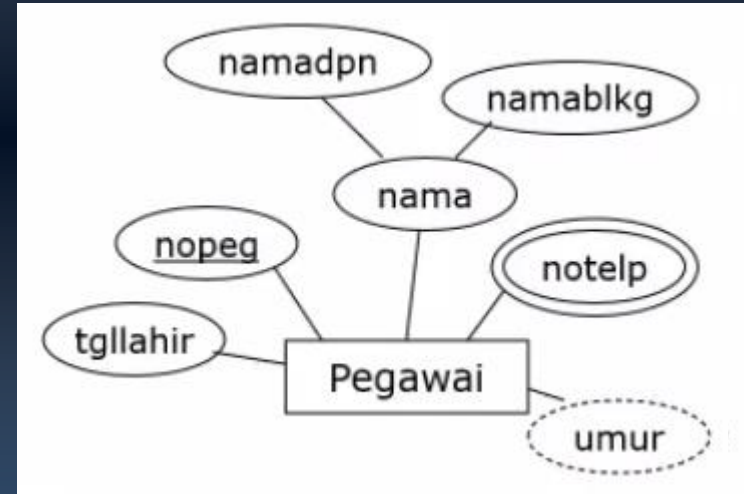
Atribut – properti dari entitas-entitas dalam himpunan itu

- ❑ Misal pegawai memiliki nopeg, nama, dan beberapa nomor telepon

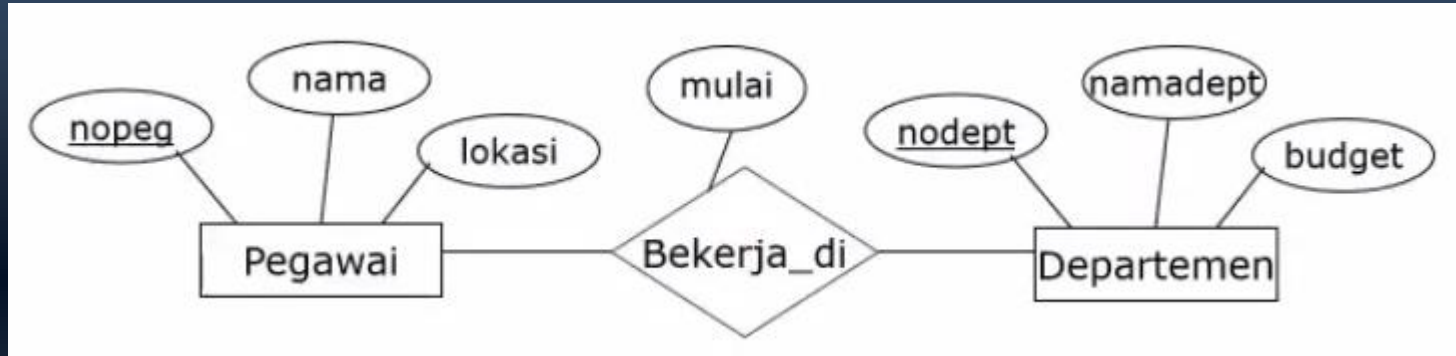
Domain – himpunan nilai yang boleh dimiliki oleh setiap atribut, misal $18 < \text{umur} < 65$

Tipe atribut:

- ❑ Simple atribut: misal nopeg, namadpn, namablkg
- ❑ Composite atribut: misal nama
- ❑ Single-valued (misal nopeg) dan multi-valued atribut misal nomor telepon
- ❑ Derived atribut, misal umur, dari tanggal lahir



ER Model - Relationship



Relationship: hubungan antara dua atau lebih himpunan entitas

❑ Misal pegawai Budi bekerja di departemen Farmasi

Untuk sebuah relationship, kita harus menentukan pegawai mana dan departemen mana yang berhubungan dalam relationship tersebut

Kita memerlukan nilai **key** seorang pegawai dan nilai key sebuah departemen, disimpan bersama-sama, untuk menunjukkan sebuah relationship

Keys



Dari perspektif basis data, setiap entitas berbeda

Key: sekelompok atribut yang dapat membedakan suatu entitas dan entitas lainnya

Step to identify primary key :

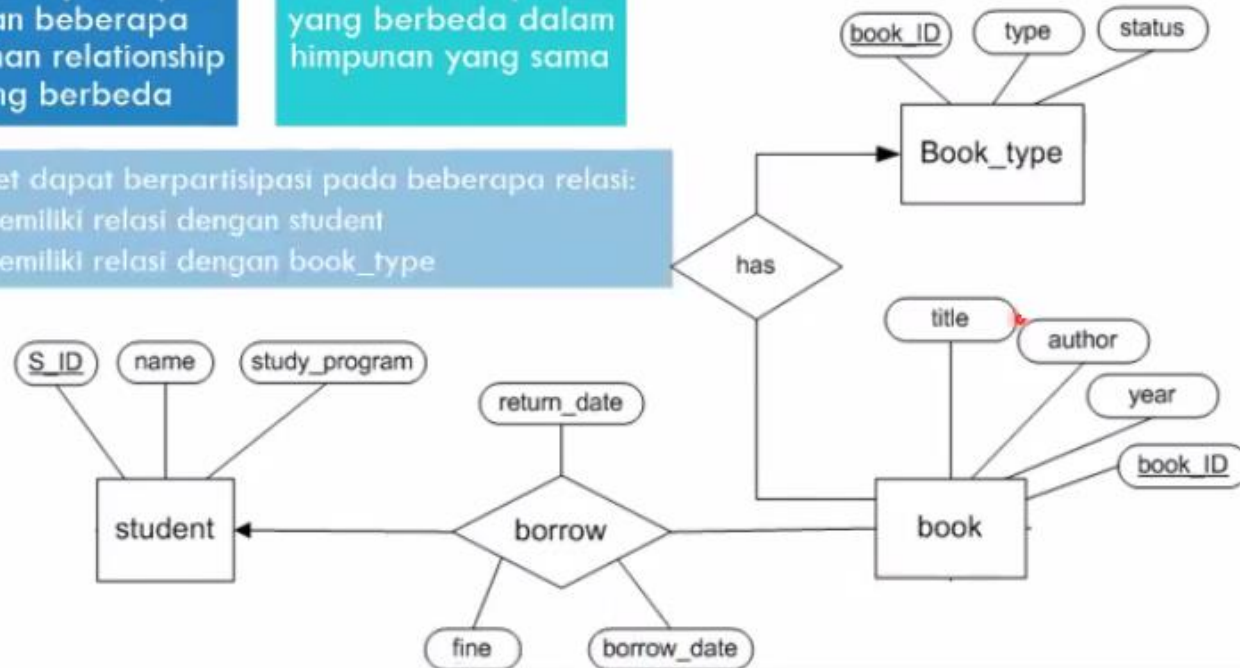
- ❑ **Super key:** satu atau lebih atribut yang nilainya dapat digunakan untuk membedakan setiap entitas
- ❑ **Candidate key:** superkey yang paling minimal
- ❑ **Primary key:** candidate key yang dipilih untuk membedakan entitas satu dan yang lainnya

Role in Relationship

Entitas yang sama dapat berpartisipasi dalam beberapa himpunan relationship yang berbeda

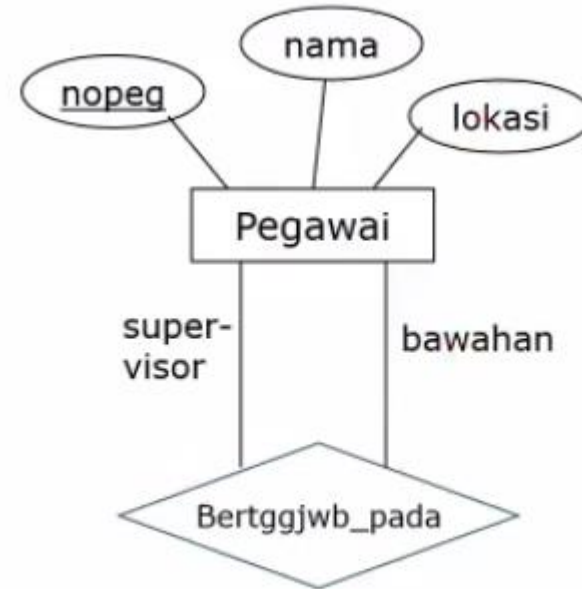
atau memiliki "peran" yang berbeda dalam himpunan yang sama

Entity set dapat berpartisipasi pada beberapa relasi:
Book memiliki relasi dengan student
Book memiliki relasi dengan book_type



Peran Dalam Relationship

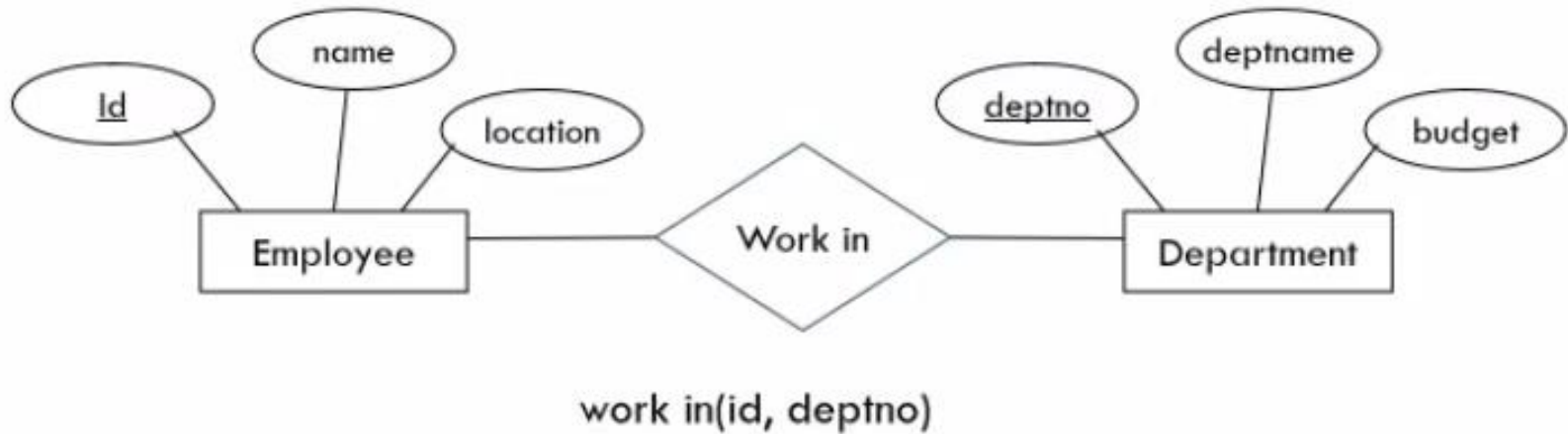
Entitas yang sama dapat memiliki “peran” yang berbeda dalam himpunan yang sama



`bertggjwb_pada(supervisor_ssn,bawahan_ssn)`

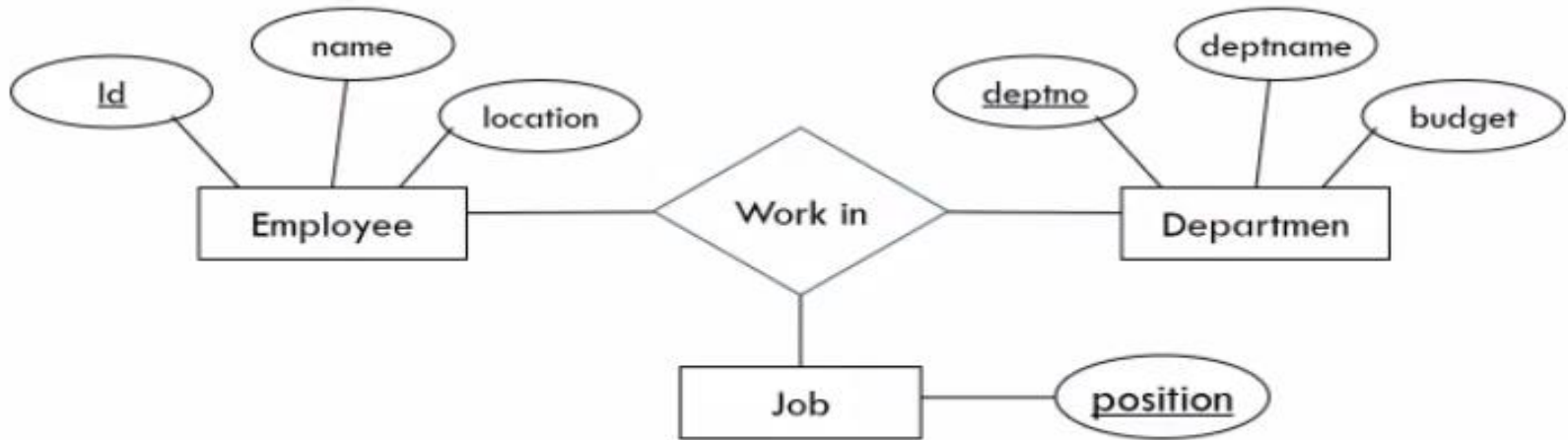
Relationship Derajat (Derajat Relasi)

Binary Relationship



Relationship Derajat (Derajat Relasi)

Ternary Relationship



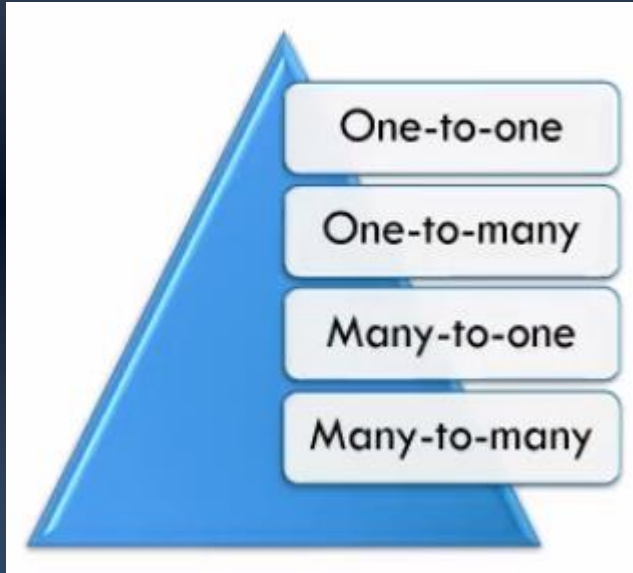
work in(id, deptno, position)

Atribut Dalam Relationship



- ❖ Sebuah himpunan relationship mungkin memiliki sebuah atribut
- ❖ Misal “mulai” menyimpan tanggal mulai bekerja seorang pegawai pada sebuah pekerjaan di sebuah departemen

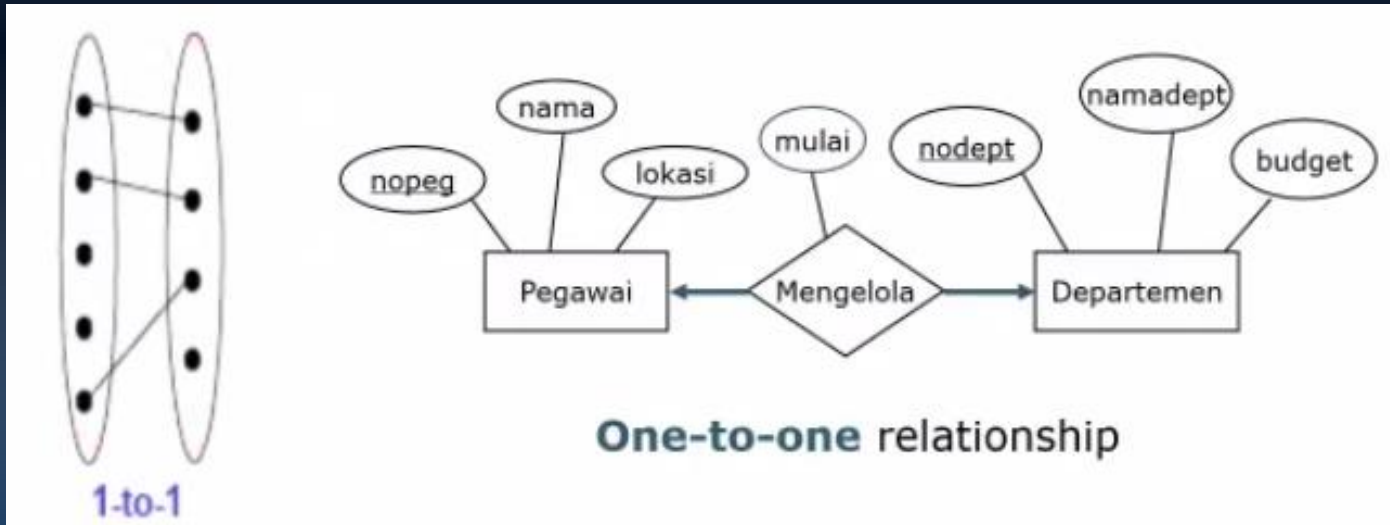
Cardinality



Cardinality menyatakan jumlah entitas yang dapat diasosiasikan dengan entitas lain menggunakan himpunan relasi

Kardinalitas Relationship

- ❖ Setiap departement memiliki *paling banyak* satu manajer, dan seorang manajer dapat mengelola *paling banyak* satu departemen

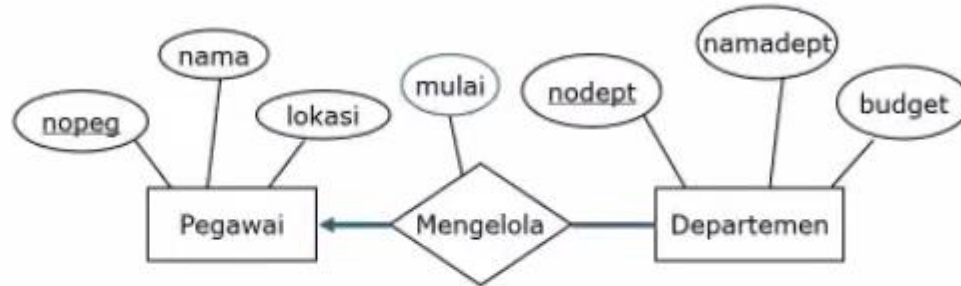


Kardinalitas Relationship

- ❖ Setiap departemen memiliki *paling banyak* satu manager, tapi seorang manager (pegawai) dapat mengelola beberapa departemen



1-to Many

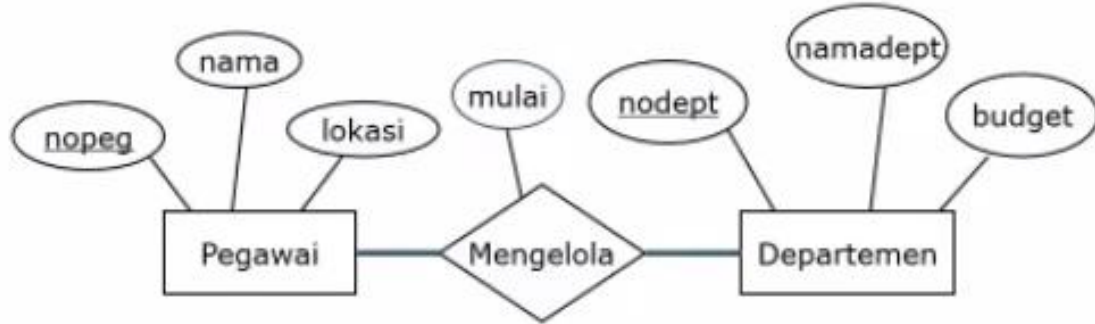
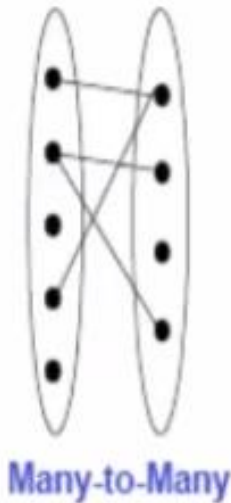


One-to-many relationship
(pegawai → departemen)

Many-to-one relationship
(departemen → pegawai)

Kardinalitas Relationship

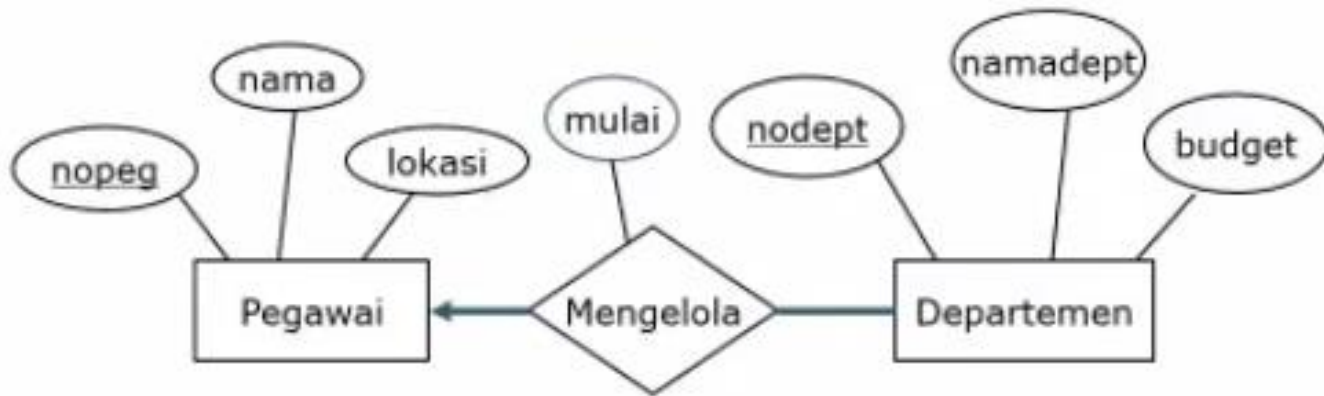
Seorang pegawai dapat bekerja di beberapa departemen, sebuah departemen dapat memiliki banyak pegawai



Many-to-many relationship

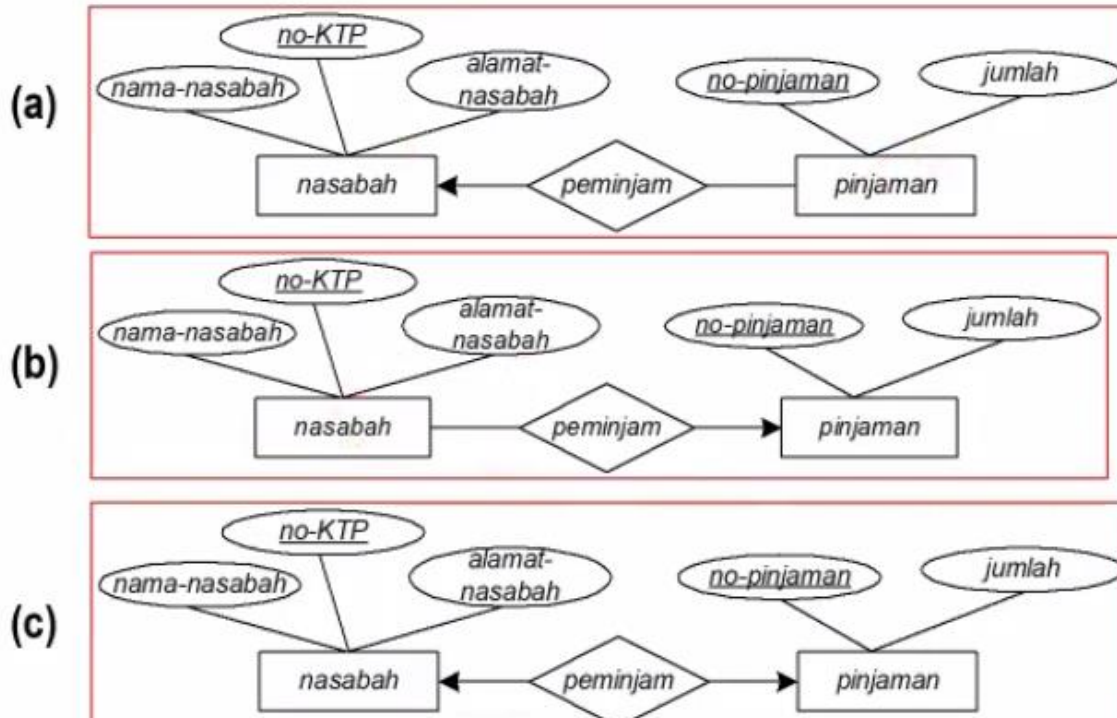
Arti dari "Tanda Panah"

- ❖ Tanda panah berarti "*paling banyak satu*"
- ❖ Tidak dapat menjamin keberadaan sebuah himpunan entitas yang ditunjuk
- ❖ Misal pada suatu saat ada departemen yang tidak memiliki manager



Arti dari "Tanda Panah"

❖ Apa perbedaan ketiga ER dibawah ini?



Participation Constraints

Partisipasi total – *total participation* (digambarkan dengan dua garis): setiap entitas dalam himpunan entitas berpartisipasi dalam minimal satu relationship di himpunan relationship

- Partisipasi dari *pinjaman* dalam *peminjaman* adalah total – setiap pinjaman pasti memiliki customer yang terhubung dengannya melalui *pinjaman*; sebuah pinjaman tidak dapat terwujud tanpa minimal satu relationship *peminjaman*

Partisipasi parsial – *partial participation*: beberapa entitas mungkin tidak berpartisipasi dalam relationship di himpunan relationship

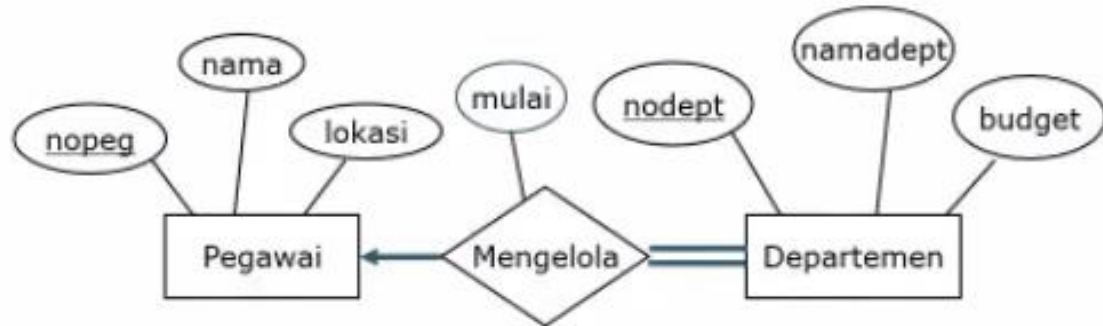
- Partisipasi dari *customer* dalam *peminjaman* adalah parsial, beberapa customer mungkin hanya punya tabungan

Participation Constraints



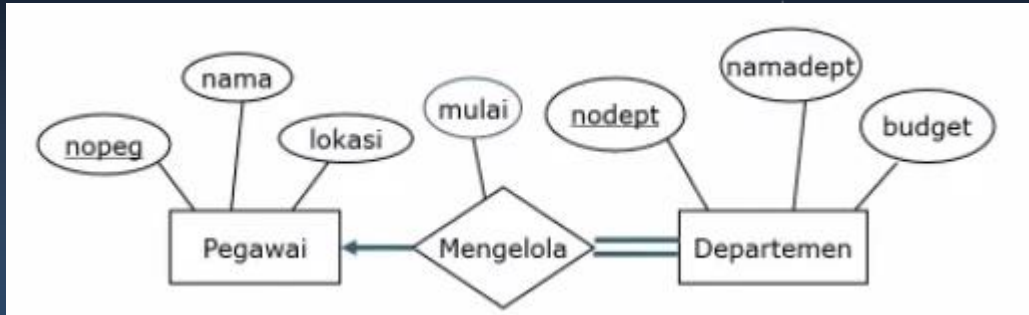
Pertanyaan

- ❖ Relationship many-to-one Mengelola menggambarkan bahwa sebuah departemen memiliki paling banyak satu manager, jadi sebuah departemen mungkin tidak memiliki manager
- ❖ Apa yang terjadi jika Departemen memiliki partisipasi total dalam Mengelola?



Referential Integrity

- ❖ Relationship many-to-one Mengelola menggambarkan bahwa sebuah departemen memiliki paling banyak satu manager, jadi sebuah departemen mungkin tidak memiliki manager
- ❖ Dikombinasikan dengan partisipasi total maka dipastikan bahwa *hanya satu* nilai yang ada



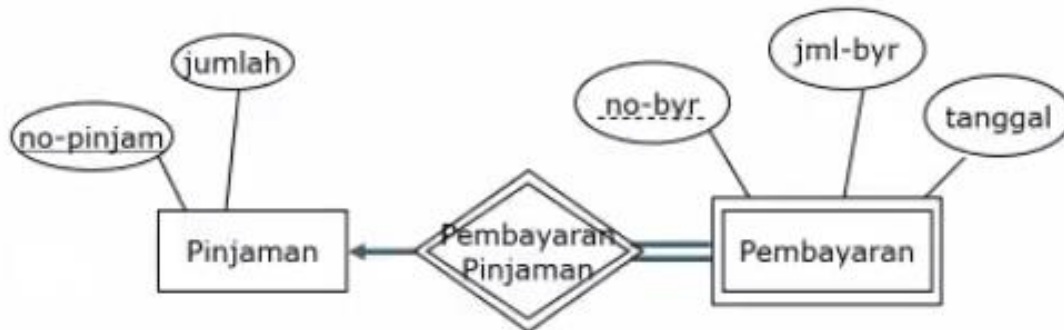
Proses penghapusan seorang pegawai yang mengelola sebuah departemen dilarang!!

Himpunan Entitas Lemah (Weak Entity)

- ❖ Himpunan entitas yang **tidak memiliki primary key**
- ❖ Keberadaan sebuah himpunan entitas lemah tergantung pada keberadaan himpunan entitas lain (*identifying entity set*)
 - ❑ Himpunan entitas lemah harus berhubungan dengan himpunan entitas lain melalui sebuah **total, one-to-many** himpunan relationship dari himpunan entitas lain ke himpunan entitas lemah
- ❖ **Pembeda (discriminator) atau partial key** dari sebuah himpunan entitas lemah adalah himpunan atribut yang membedakan antara semua entitas dalam himpunan entitas lemah
- ❖ **Primary key** dari himpunan entitas lemah adalah primary key dari himpunan entitas kuat ditambah dengan discriminator dari himpunan set entitas lemah

Contoh

- ❖ Himpunan entitas lemah digambarkan dengan persegi panjang doppel
- ❖ Discriminator diberi garis bawah putus-putus
- ❖ *no-pembayaran* – discriminator himpunan entitas *pembayaran*
- ❖ Apa primary key untuk *pembayaran*?
 - ❖ (no-pinjam,no-byr)



Himpunan Entitas Kuat VS Lemah

Himpunan entitas kuat:

- ❑ Memiliki cukup atribut untuk membentuk primary key

Himpunan entitas lemah:

- ❑ Tidak memiliki cukup atribut untuk membentuk primary key
- ❑ Jadi, tidak cukup atribut untuk membentuk key *apapun*

Tapi setiap himpunan entitas memerlukan key; Jadi?

- ❑ Harus *mengimpor* atribut dari himpunan entitas kuat



References

- Database System Concept Silberschatz Slides-<http://codex.cs.yale.edu/avi/db-book/db6/slide-dir/index.html>
- Slide Relational Database, Hilda Widyastuti
- Slide Basis Data, Metta Santiputri

+++

Thanks!

Do you have any questions?

