

# Perancangan Basis Data

## Pertemuan 4

Transformasi Rule ke Cardinality Diagram ER, dan Transformasi Diagram ER ke Tabel

# Tujuan Pertemuan

- Mahasiswa akan mampu menganalisis aturan dan melakukan transformasi aturan menjadi cardinality pada diagram ER.
- Mahasiswa akan mampu menentukan jenis suatu binary relationship berdasarkan cardinality tiap entity.
- Mahasiswa akan mampu menganalisis ER dan melakukan transformasi ER menjadi tabel, dengan memperhatikan cardinality entitas dan mengikuti aturan-aturan transformasi.
- Mahasiswa akan mampu menetapkan cardinality atribut.

# Cardinality Diagram ER (Transformasi Rule ke Cardinality ERD)

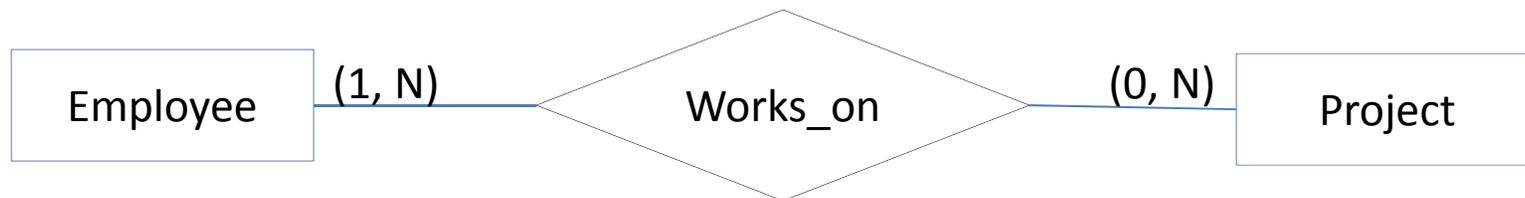
## Rule:

- Setiap employee harus bekerja (work on) pada minimal satu project dan mungkin bekerja pada beberapa project;

**Card(Employee, works\_on) = (1, N)**

- Suatu project mungkin tidak memiliki employee untuk periode tertentu (menunggu penugasan employee), dan beberapa project akan memiliki beberapa employee yang bekerja pada project tersebut;

**Card(Project, works\_on) = (0, N)**



# Cardinality Diagram ER (Transformasi Rule ke Cardinality ERD)

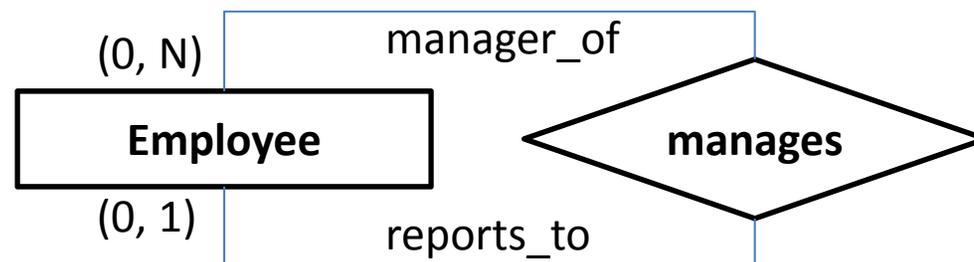
## Rule:

- Employee yang menjadi manager (manager\_of) mungkin tidak memiliki anak buah untuk waktu tertentu dan tetap disebut manager;

**Card(Employee(manager\_of), manages) = (0, N)**

- Seorang employee membuat laporan ke (reports\_to) hanya pada satu manager, tetapi dimungkinkan juga seorang employee tidak membuat laporan kepada siapapun (hal ini karena pasti ada employee yang memiliki tingkatan tertinggi dalam hirarki yang tidak memiliki manager/top level management/big boss).

**Card(Employee(reports\_to), manages) = (0, 1)**



# Cardinality Diagram ER (Transformasi Rule ke Cardinality ERD)

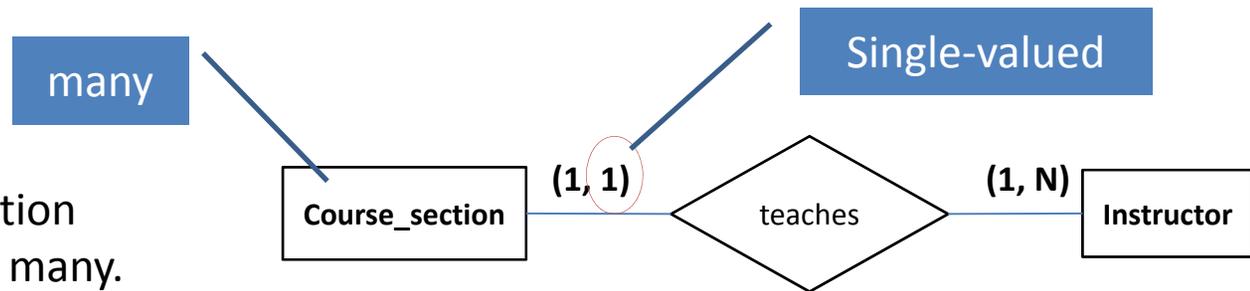
## Definition.

1. Suatu entitas **E** terhubung pada relationship **R** dengan **max-card(E, R) = 1**, maka E dikatakan memiliki partisipasi **singel valued** dalam relationship R. Jika **max-card(E, R) = N**, maka E dikatakan memiliki partisipasi **multivalued** dalam relationship R.
2. Suatu binary relationship R antara entitas E dan F dikatakan:
  - **many-to-many**, or N-N, jika Kedua entitas E and F adalah **multi-valued** dalam relationship.
  - **one-to-one**, or 1-1, jika Kedua entitas E dan F adalah **single-valued** dalam relationship.
  - **many-to-one**, or N-1. (1-N relationship sama N-1 relationship.), jika E adalah **single-valued** dan F adalah **multivalued**, atau sebaliknya.

# One-to-One, Many-to-Many, and Many-to-One Relationships

- Pada binary **relationship many-to-one** sisi “**many**” berada pada sisi yang memiliki partisipasi **single-valued**.

Contoh:



Karena **Course\_section** merupakan entitas **many**, maka:

**lid** yang merupakan atribut relasi, mungkin muncul lebih dari satu pada entitas **Course\_section**

<u>cid</u>	co_name	iid	<u>iid</u>	i_name
C01	Database 1	001	001	Budi
C02	Database 2	001	002	Ani
C03	Analysis	002	003	Dedi

# One-to-One, Many-to-Many, and Many-to-One Relationships

## Definisi.

- Ketika entitas E yang berpartisipasi dalam suatu relationship dengan **min-card(E, R) = 1**, dikatakan memiliki partisipasi **mandatory** (wajib) dalam R, sebaliknya **min-card(E, R) = 0**, disebut optional atau partisipasi **optional** (opsional).

# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 3:



**N – N Relationships** (many-to-many):

Ketika dua entitas E dan F memiliki relasi binary many-to-many pada Relationship R, **Relationship R harus dipetakan menjadi tabel R** dalam disain relational database.

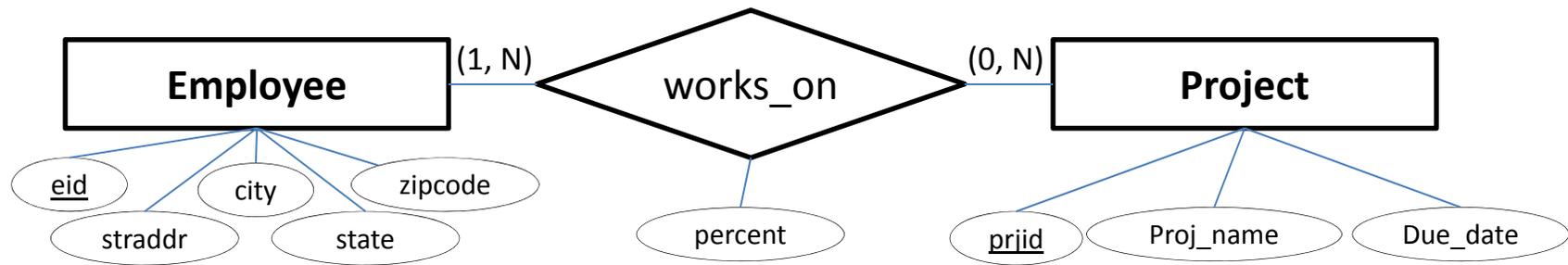
Tabel R tersebut memiliki kolom yang menampung **semua atribut primary key** (kunci utama) **dari dua tabel yang terelasi** (tabel E dan F) dan **kombinasi kolom ini membentuk primary key** (kunci utama) untuk tabel R.

Tabel R juga berisi kolom untuk semua atribut yang dimilikinya (atribut pada relationship).

Data kejadian/fakta relationship yang diwakili oleh deretan baris pada tabel, disertai dengan instance (atribut primary key dari entitas) diidentifikasi melalui kunci utama mereka sebagai baris.

# Transformasi Binary Relationship menjadi tabel (Relations)

## Contoh Transformation Rule 3:



**employee**

<u>eid</u>	straddr	city	state	zipcode
197	7 Beacon St	Boston	MA	02102
221	19 Brighton St	Boston	MA	02103
303	153 Mass Ave	Cambridge	MA	02123
197	7 Beacon St	Boston	MA	02102
...	...	...	...	...

**works\_on**

<u>eid</u>	<u>prjid</u>	percent
197	p11	50
197	p13	25
197	p21	25
221	p21	100
303	p13	40
303	p21	60
...	...	...

**project**

<u>prjid</u>	proj_name	due_date
p11	Phoenix	3/31/99
p13	Excelsior	9/31/99
p21	White Mouse	6/30/00
...	...	...

# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 4:



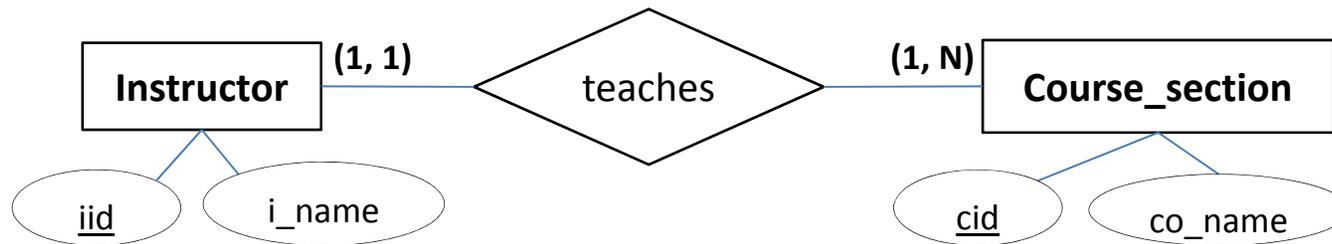
### N – 1 Relationships (many-to-one):

- Ketika dua entitas E dan F memiliki relasi binary many-to-one pada Relationship R, **relationship tidak akan dipetakan menjadi tabel** dalam disain relational database.
- Jika diasumsikan entitas F memiliki **max-card(F, R) = 1** (F sebagai entitas “many”) tabel F (hasil transformasi entitas F) **harus menyertakan primary key (kunci utama)** tabel E (hasil transformasi entitas E), yang selanjutnya disebut **foreign key** pada tabel F.
- Karena  $\text{max-card}(F, R)=1$ , maka tiap row pada tabel F ter-relasi melalui nilai foreign key dengan (maksimum satu) nilai primary key pada tabel E.
  - Jika F memiliki partisipasi **mandatory (min-card(F, R)=1)**, maka nilai foreign key pada tiap row tabel F harus terrelasi secara tepat pada satu nilai primary key pada tabel E, artinya bahwa foreign key pada tabel F **tidak boleh bernilai null**.
  - Jika F memiliki partisipasi **optional (min-card(F, R)=0)**, maka nilai foreign key pada tiap row tabel F **boleh bernilai null**.



# Transformasi Binary Relationship menjadi tabel (Relations)

## Contoh Transformation Rule 4:



**instructor**

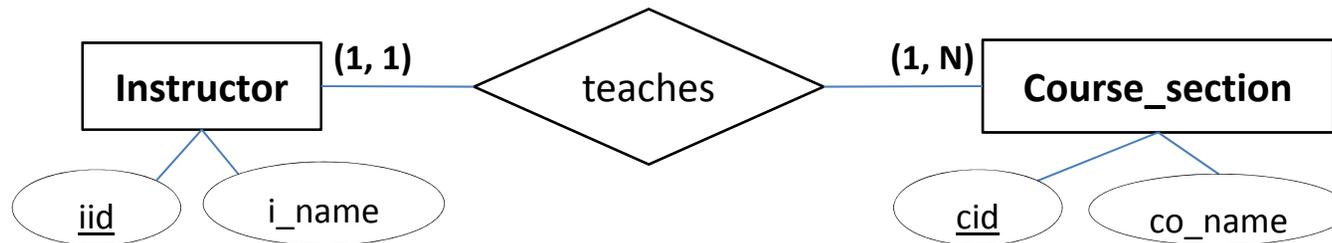
<u>iid</u>	i_name
001	Budi
002	Ani
003	Dedi

**course\_section**

<u>cid</u>	co_name	iid
C01	Database 1	001
C02	Database 2	001
C03	Analysis	002

# Transformasi Binary Relationship menjadi tabel (Relations)

## Kasus Transformation Rule 4



- Satu instructor dapat mengajar (teach) beberapa course section, tetapi tiap course section hanya bisa dan harus memiliki satu instructor. (**mandatory**).
- Instruksi create table pada SQL memungkinkan untuk **mengatur agar suatu kolom not null** (tidak boleh berisi nilai null);

```
CREATE TABLE course_section(cid char(3) NOT NULL, co_name varchar(50), iid char(3) NOT NULL, PRIMARY KEY (cid));
```

hal ini memungkinkan kita untuk memastikan mandatory diterapkan pada tabel hasil transformasi dari entitas “many” dalam many-to-one relationship.

Dengan demikian kita bisa membuat tabel **course\_section** dengan kolom **foreign key (iid) not null** (tidak boleh null) saat suatu row di-**insert** pada tabel **course\_section**.

# Transformasi Binary Relationship menjadi tabel (Relations)

## Kasus Transformation Rule 4

instructor

<u>iid</u>	i_name
001	Budi
002	Ani
003	Dedi

course\_section

<u>cid</u>	co_name	iid
C01	Database 1	001
C02	Database 2	001
C03	Analysis	002

- Kolom iid pada tabel course\_section adalah foreign key, yang me-relasikan course\_section dengan instructor pada tabel instructor.
- Jika user melakukan **update** data kolom iid pada tabel instructor, berarti terjadi **perbedaan nilai** antara kolom **iid pada tabel instructor** dengan **iid pada tabel course\_section**, hal ini menjadi **constraint** (kendala) pada integritas relationship.
- Untuk menerapkan constraint ini bisa digunakan **constraint** pada SQL. Constraint bisa digunakan untuk memberlakukan bahwa nilai kolom iid (foreign key) pada tabel course\_section (**tabel anak/sisi "many"**) benar-benar ada sebagai nilai dalam kolom iid (primary key) pada tabel instructor (**tabel induk/sisi "one"**).

```
CREATE TABLE course_section(cid char(3) NOT NULL, co_name varchar(50), iid char(3) NOT NULL, PRIMARY KEY (cid), constraint foreign key (iid) references instructor(iid) );
```

# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 5:



### 1-1 Relationships, Optional Participation (one-to-one):

- Ketika dua entitas E dan F memiliki relasi binary one-to-one pada Relationship R, **relationship R tidak akan dipetakan menjadi tabel** dalam disain relational database.
- Jika salah satu entitas (E) memiliki partisipasi optional (min-card(E, R)=0), transformasi mengacu pada **transformation rule 1**, dan pada **tabel E** (yang memiliki min-card(E, R)=0) harus ditambahkan kolom yang berisi atribut primary key tabel F (yang memiliki min-card(F, R)=1).
- Kolom pada tabel E tersebut disebut **foreign key**, dan **sebagai penghubung yang merujuk kepada kolom primary key pada tabel F** (tabel rujukan).

# Transformasi Binary Relationship menjadi tabel (Relations)

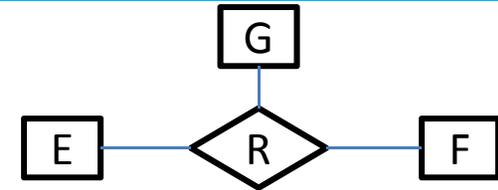
## Transformation Rule 6:



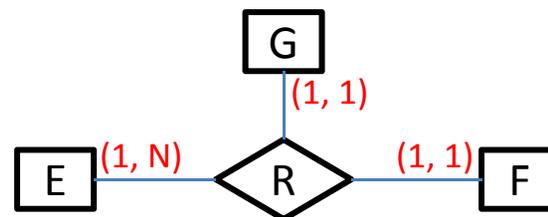
**1-1 Relationships, Mandatory Participation pada kedua entitas (one-to-one):**

- Dalam kasus relationship one-to-one dengan partisipasi mandatory (wajib) pada kedua entitas,
- Cara yang paling tepat untuk kasus ini adalah dengan **mengkombinasikan tabel dari dua entitas tersebut dan menggabungkannya menjadi satu tabel,**
- Cara ini mengantisipasi masalah penentuan tabel mana yang akan ditambahkan foreign key.

# Transformasi n-ary Relationship menjadi tabel (Relations)



- Sejauh ini belum ada transformation rules untuk relationship n-ary dimana  $n > 2$ .
- Tetapi, jika **satu entitas memiliki max-card=N** dan **entitas lainnya memiliki max-card=1**, maka; relationship dapat direpresentasikan dengan n-1 foreign key merujuk ke satu table yang memiliki max-card=N

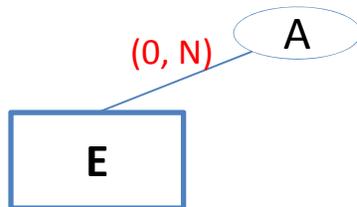


# ER Concept Tambahan

## Attribute Cardinality

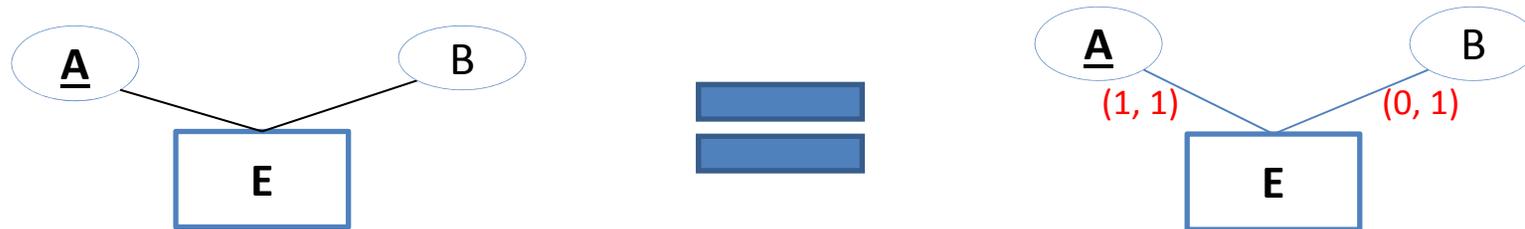
A

- Misal, Suatu entitas E dengan atribut A, maka; 
  - $\text{min-card}(A, E) = 0 \rightarrow$  mengindikasikan A adalah **optional**, dan
  - $\text{min-card}(A, E) = 1 \rightarrow$  mengindikasikan A adalah **mandatory**, maka kolom A adalah **NOT NULL**
  - $\text{max-card}(A, E) = 1 \rightarrow$  mengindikasikan atribut A adalah **single valued**, dan
  - $\text{max-card}(A, E) = N \rightarrow$  mengindikasikan atribut A adalah **multivalued**.
  - $\text{Card}(A, E) = (0, N) \rightarrow \text{min-card}(A, E)=0, \text{max-card}(A, E) = N$



# ER Concept Tambahan

## Attribute Cardinality

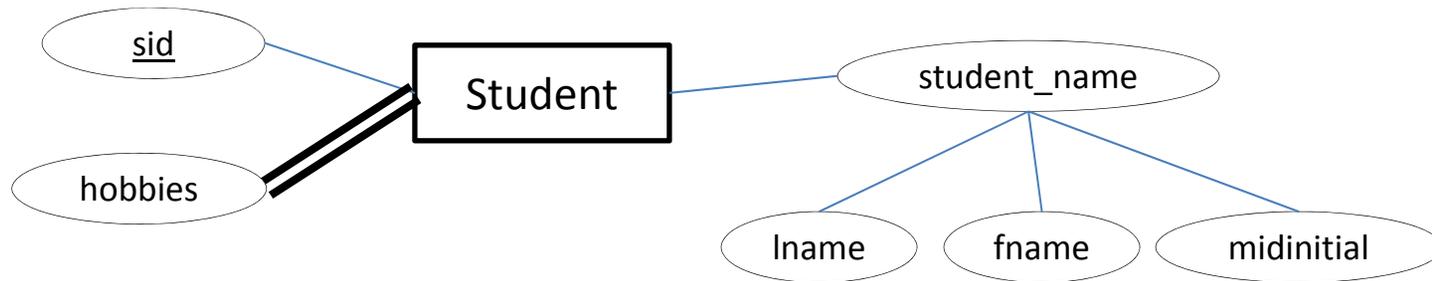


- Jika suatu konektor atribut tidak memiliki label cardinality, maka;
  - Jika atribut tersebut adalah **identifier** → **cardinality= (1, 1)**
  - Jika atribut tersebut adalah **descriptor** → **cardinality= (0, 1)**

# ER Concept Tambahan

## QUIZ (**Attribute Cardinality**)

- Buatlah attribute Cardinality untuk diagram ER Berikut:



# ER Concept Tambahan

## Weak Entity

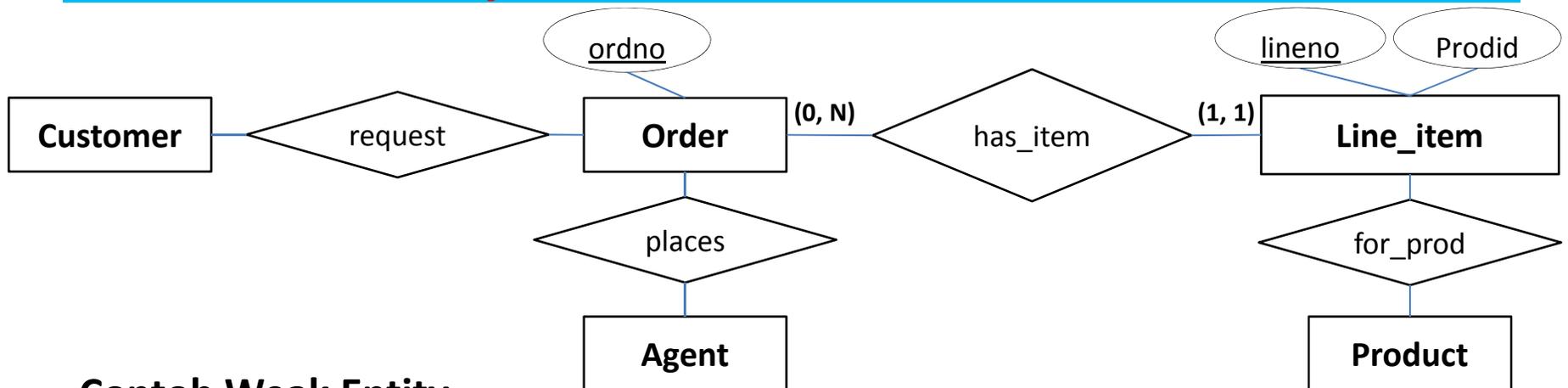
### Definition:

- Weak Entity:

Entitas yang kejadiannya tergantung pada keberadaannya melalui suatu relationship R, pada kejadian entitas (strong) lainnya.

# ER Concept Tambahan

## Weak Entity



### Contoh Weak Entity

- entitas Order adalah **optional** pada relationshipnya ke Line\_item, karena order tetap bisa mulai walau tanpa line item sekalipun.
- Entitas Line\_item adalah **mandatory** pada relationshipnya ke Order, karena line-item order (daftar pesanan) untuk suatu product tidak mungkin ada jika pada master order tidak dimasukkan product tersebut sebagai product yang dipesan customer pada pesanan. Dengan demikian maka **entitas Line\_item** adalah **weak entity**.
- Jika pesanan terhadap product tersebut di cancel oleh customer, maka semua kejadian pada Line\_item (weak entity) tersebut juga akan hilang.

# ER Concept Tambahan

## Weak Entity

- **Transformasi Weak Entity ke Tabel:**

Ketika Line\_item (weak entity) dipetakan ke tabel line\_item, maka **kolom ordno ditambahkan ke tabel line\_item** (mengikuti **transformation rule 4**) untuk merepresentasikan relasi item, dan **primary key tabel line\_item** adalah **kombinasi** dari ordno dan lineno.

order

<u>ordno</u>	custid
ord001	cst001
ord002	cst002
ord003	cst001
ord004	cst003
...	...

line\_item

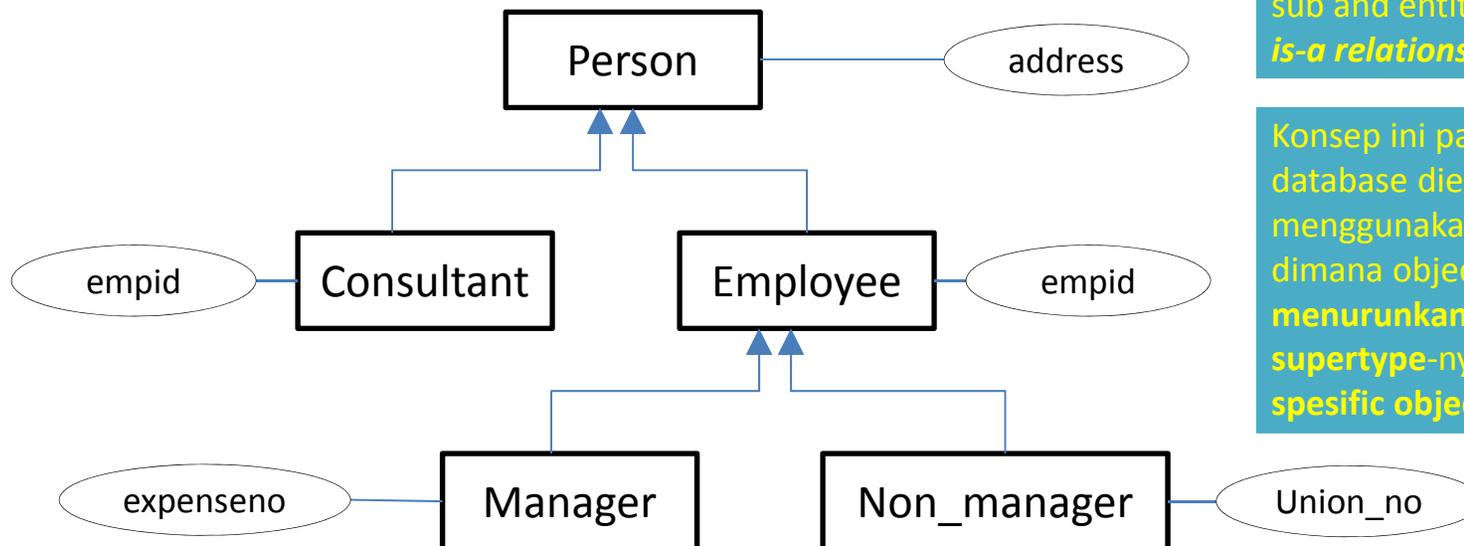
<u>lineno</u>	<u>ordno</u>	prodid
001	ord001	prd001
002	ord001	prd002
001	ord002	prd002
002	ord002	prd004
...	...	...

# ER Concept Tambahan

## Generalization Hierarchies

- **Konsep generalization hierarchy or generalization relationship**

- Ide dari generalization relationship adalah; beberapa entitas yang memiliki atribut yang sama (entitas general) dapat digeneralisasi menjadi entitas super (pada level yang lebih tinggi),
- atau sebagai alternatif, entitas general dapat diuraikan (dekomposisi) menjadi entitas sub (pada level dibawahnya).
- Tujuannya adalah untuk menempelkan atribut kepada level yang tepat, dengan demikian entitas terhindar dari memiliki atribut umum yang kemungkinan memiliki banyak nilai null



The arrow relationship antara entitas sub and entitas supertype disebut; **is-a relationship**

Konsep ini pada Object-relational database diekspresikan menggunakan tipe **inheritance**, dimana object (rows) pada subtype **menurunkan (inherit) atribut supertype-nya** dan **memiliki atribut spesifik object.**

# See You Next Session

- **Thank's**

# Rangkuman Transformasi ERD ke Tabel

# Intro ER Concept

## -Transformasi ER Menjadi Tabel

Transformasi entity atau relasi menjadi tabel relasi.

Terdapat dua aturan (*rule*) dalam melakukan transformasi ini:

- **Transformation Rule 1:**

- Setiap **entity/relasi** dalam ER dipetakan ke/menjadi **tabel**. Satu entity/relasi adalah satu tabel relasi, dan beri nama tabel tersebut sesuai dengan nama entity/relasi-nya.
- **Kolom** pada tabel merupakan **representasi dari atribut yang bukan multivalue**, atribut **subset dari atribut komposit** (catatan: atribut komposit-nya sendiri tidak dibuatkan kolom).
- Identifier suatu Entity dipetakan menjadi candidate key pada tabel, dan **primary identifier** dipetakan menjadi **primary key**.

*Catatan: primary identifier bisa saja berupa atribut komposit, jika hal tersebut terjadi, maka; semua atribut subset dari atribut komposit tersebut dipetakan menjadi primary key.*

- Data kejadian dari entity kemudian dipetakan ke dalam tabel berupa baris (row) pada tabel.

# Intro ER Concept

## -Transformasi ER Menjadi Tabel

- **Transformation Rule 2:** (atribut multivalued menjadi entity)
  - Jika pada entity terdapat atribut multivalued, buatlah entity baru untuk menampung **atribut multivalued (a)**, juga sertakan atribut **primary identifier dari entity asal (p)** pada entity baru tersebut.
  - Entity baru tersebut diberi **nama yang sama dengan nama atribut multivalued**.
  - Petakan entity baru tersebut menjadi tabel dengan nama yang sama dengan nama entity-nya.
  - **Kolom** pada tabel merupakan representasi dari **p** yang diikuti dengan **a** secara berurutan.
  - **Primary key dari tabel** ini adalah kolom **p** dan kolom **a**.
  - Data kejadian dari entity kemudian dipetakan ke dalam tabel berupa baris (row) tabel yang sesuai pasangannya (**p,a**).

# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 3:



**N – N Relationships** (many-to-many):

Ketika dua entitas E dan F memiliki relasi binary many-to-many pada Relationship R, **Relationship R harus dipetakan menjadi tabel R** dalam disain relational database.

Tabel R tersebut memiliki kolom yang menampung **semua atribut primary key** (kunci utama) **dari dua tabel yang terelasi** (tabel E dan F) dan **kombinasi kolom ini membentuk primary key** (kunci utama) untuk tabel R.

Tabel R juga berisi kolom untuk semua atribut yang dimilikinya (atribut pada relationship).

Data kejadian/fakta relationship yang diwakili oleh deretan baris pada tabel, disertai dengan instance (atribut primary key dari entitas) diidentifikasi melalui kunci utama mereka sebagai baris.

# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 4:



### N – 1 Relationships (many-to-one):

- Ketika dua entitas E dan F memiliki relasi binary many-to-one pada Relationship R, **relationship tidak akan dipetakan menjadi tabel** dalam disain relational database.
- Jika diasumsikan entitas F memiliki **max-card(F, R) = 1** (F sebagai entitas “many”) tabel F (hasil transformasi entitas F) **harus menyertakan primary key (kunci utama)** tabel E (hasil transformasi entitas E), yang selanjutnya disebut **foreign key** pada tabel F.
- Karena  $\text{max-card}(F, R)=1$ , maka tiap row pada tabel F ter-relasi melalui nilai foreign key dengan (maksimum satu) nilai primary key pada tabel E.
  - Jika F memiliki partisipasi **mandatory (min-card(F, R)=1)**, maka nilai foreign key pada tiap row tabel F harus terrelasi secara tepat pada satu nilai primary key pada tabel E, artinya bahwa foreign key pada tabel F **tidak boleh bernilai null**.
  - Jika F memiliki partisipasi **optional (min-card(F, R)=0)**, maka nilai foreign key pada tiap row tabel F **boleh bernilai null**.



# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 5:



### 1-1 Relationships, Optional Participation (one-to-one):

- Ketika dua entitas E dan F memiliki relasi binary one-to-one pada Relationship R, **relationship R tidak akan dipetakan menjadi tabel** dalam disain relational database.
- Jika salah satu entitas (E) memiliki partisipasi optional ( $\text{min-card}(E, R)=0$ ), transformasi mengacu pada **transformation rule 1**, dan pada **tabel E** (yang memiliki  $\text{min-card}(E, R)=0$ ) harus ditambahkan kolom yang berisi atribut primary key tabel F (yang memiliki  $\text{min-card}(F, R)=1$ ).
- Kolom pada tabel E tersebut disebut **foreign key**, dan **sebagai penghubung yang merujuk kepada kolom primary key pada tabel F** (tabel rujukan).

# Transformasi Binary Relationship menjadi tabel (Relations)

## Transformation Rule 6:



**1-1 Relationships, Mandatory Participation pada kedua entitas (one-to-one):**

- Dalam kasus relationship one-to-one dengan partisipasi mandatory (wajib) pada kedua entitas,
- Cara yang paling tepat untuk kasus ini adalah dengan **mengkombinasikan tabel dari dua entitas tersebut dan menggabungkannya menjadi satu tabel,**
- Cara ini mengantisipasi masalah penentuan tabel mana yang akan ditambahkan foreign key.

Done