Perancangan Basis Data

Pertemuan 3

ER Concept Lanjutan

Tujuan Pertemuan

- Mahasiswa akan mampu menjelaskan konsep dasar relationship antara entity.
- Mahasiswa akan mampu menjelaskan konsep Cardinality (kardinalitas) sebagai rule untuk menjaga relationship pada normal database.
- Mahasiswa akan mampu mebuat diagram ER lengkap dengan cardinality-nya (sebagai rule untuk menjaga relationship pada normal database).

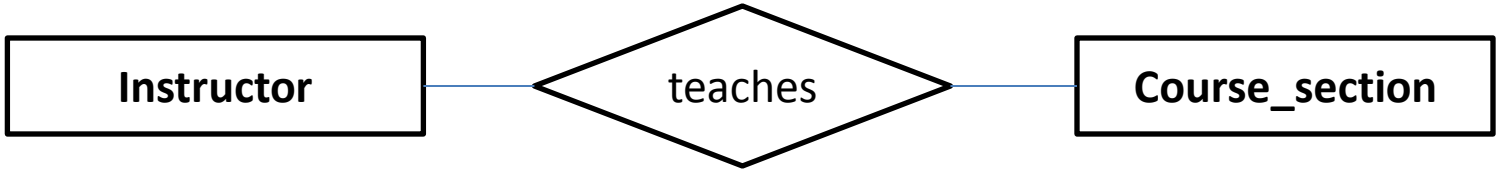
ER Concept (Relationships Among Entities)

- **Relationship**: hubungan atau interaksi antara satu entity dengan entity lainnya.

- **Degree of the Relationship** ditentukan berdasarkan jumlah entity yang terhubung pada suatu relationship
  - **Binary Relationship**: relationship antara 2 entity
  - **Ternary Relationship**: relationship dari 3 entity
  - **n-ary Relationship**: relationship dengan entity yang lebih dari 2

ER Concept (Relationships Among Entities)

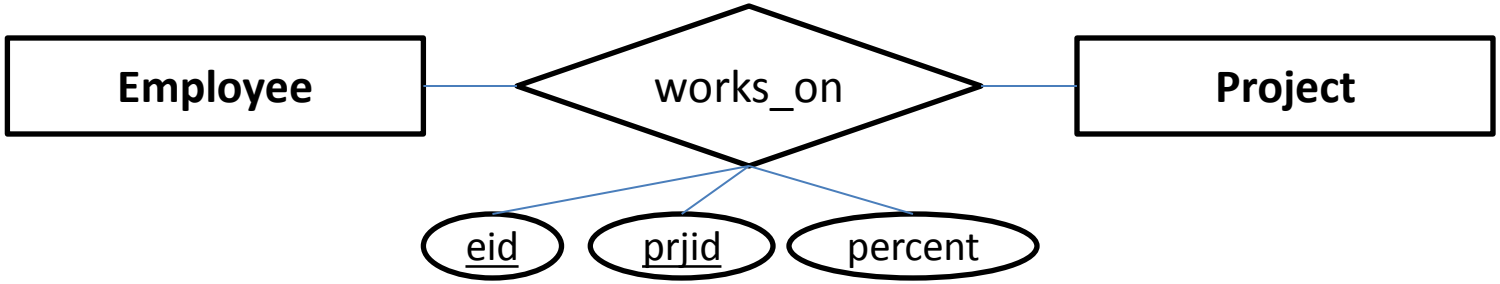
Contoh (binary relationship):



- teaches merupakan **binary relationship** antara **Instructor** dan **Course_section** .
- Relationship teaches dibentuk karena instructor tertentu mengajarkan kursus tertentu.

ER Concept (Relationships Among Entities)

Contoh (relationship dengan atribut):



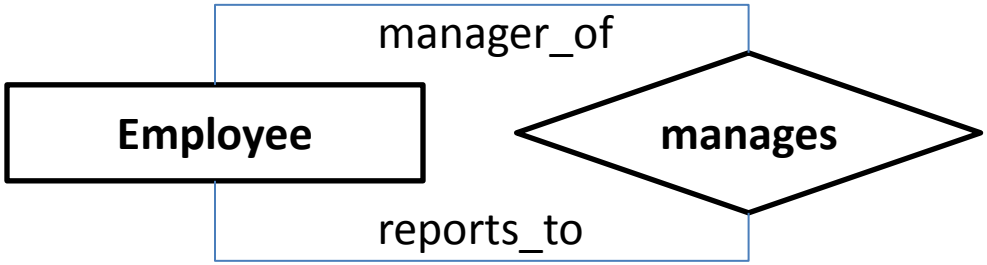
- works_on merupakan **binary relationship** antara **Employee** dan **Project**.

- Relationship works_on memiliki attribute **percent** karena **Employee** tertentu ditugaskan bekerja pada **Project** tertentu dengan persentase waktu kerja tertentu dalam seminggu.

  (**percent** = persentase waktu Employee bekerja di project terhadap waktu kerja karyawan dalam seminggu).

ER Concept (Relationships Among Entities)

Contoh (ring relationship):

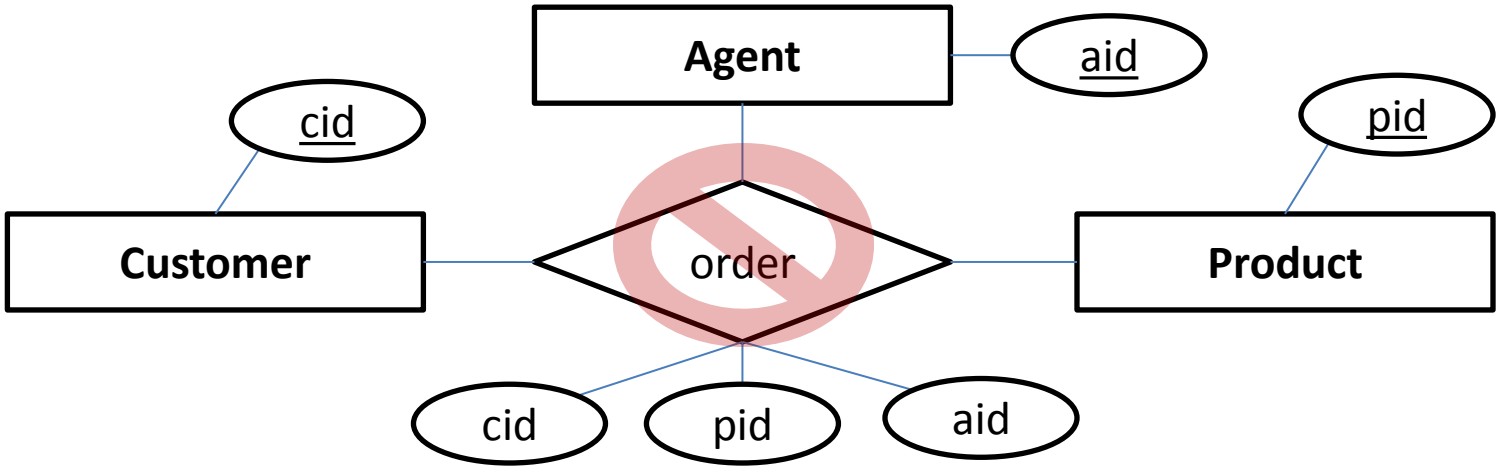


- **manages** merupakan **binary relationship** antara Employee dan Employee.
- Relationship seperti ini dinamakan **recursive relationship** atau **ring**.
- Pada relationship model ring, garis penghubung biasanya di beri **label** dengan nama sesuai peran garis tersebut.
- Ilustrasi;

  satu Employee menjadi manager untuk Employee lainnya.

ER Concept (Relationships Among Entities)

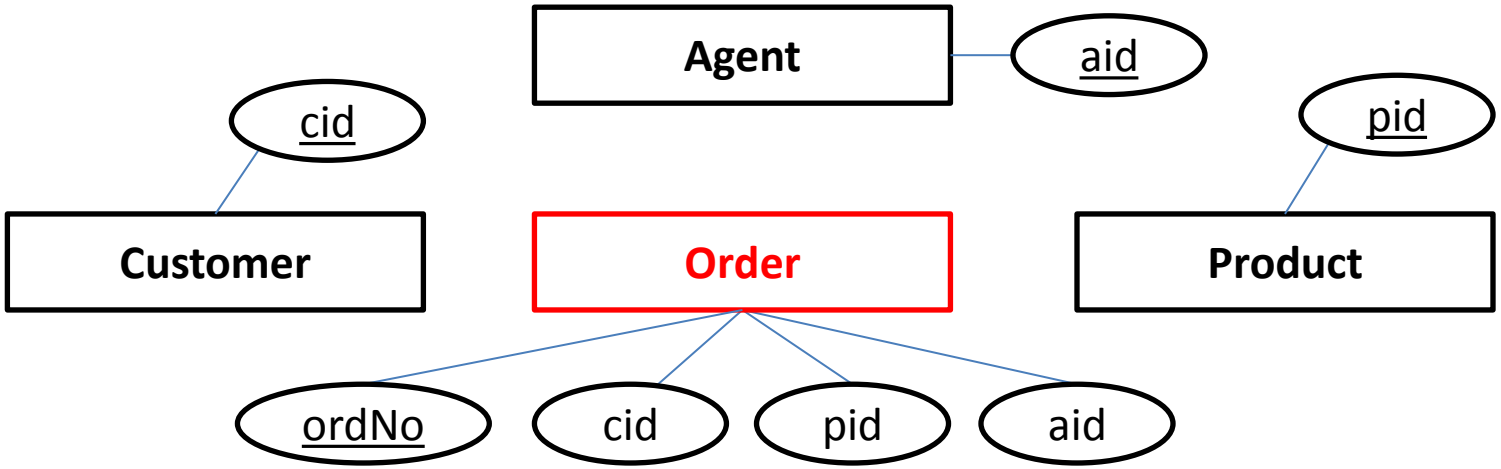
Contoh (relationship yang salah):



- Tabel order adalah **bukan relationship** terhadap Customer, Agent dan Product, karena kombinasi id (**cid**, **aid**, **pid**) tidak bisa dijadikan identifier.
- Nilai **cid**, **aid** dan **pid** yang sama dapat muncul lebih dari sekali, atau customer (cid) yang sama bisa order product (pid) yang sama ke agent (aid) yang sama.

ER Concept (Relationships Among Entities)

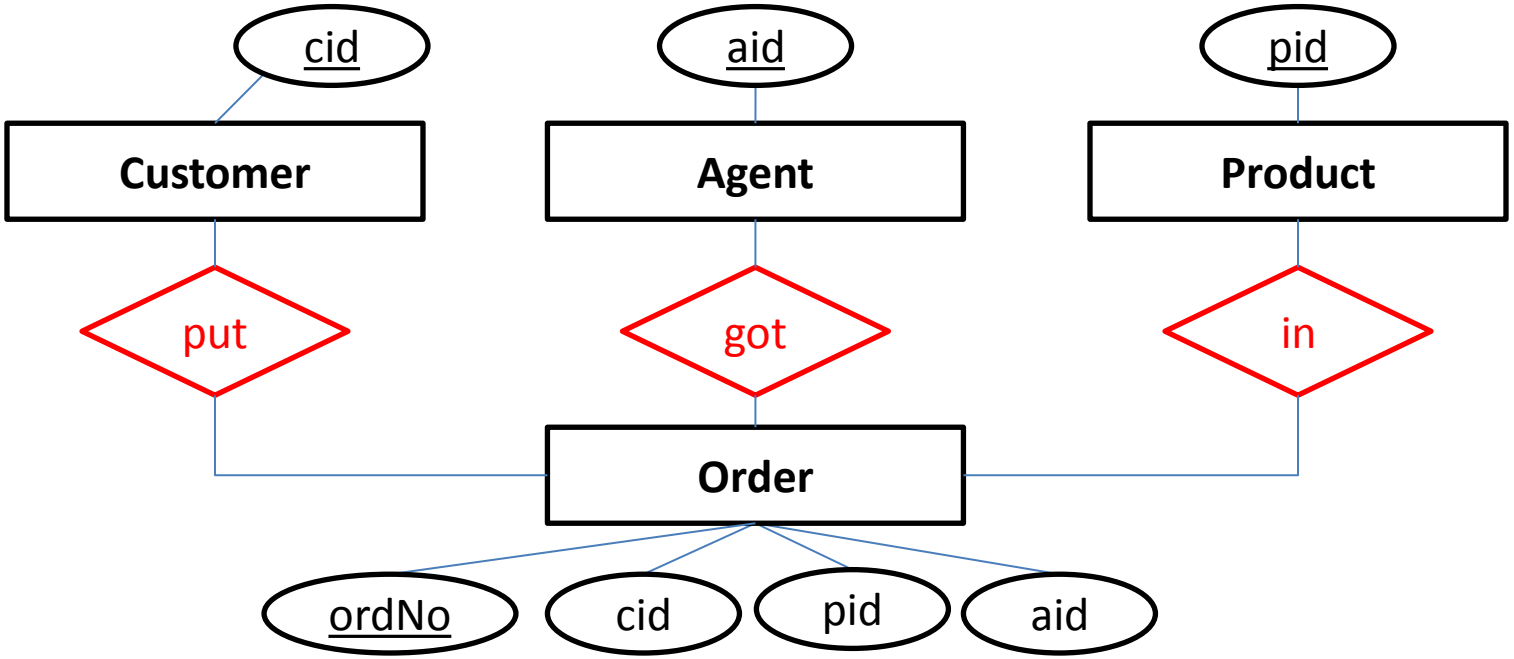
Contoh (perbaikan relationship yang salah):



- Dengan demikian tabel order akan lebih tepat sebagai **entity** dengan atribut identifier **ordNo**, daripada sebagai relationship.

ER Concept (Relationships Among Entities)

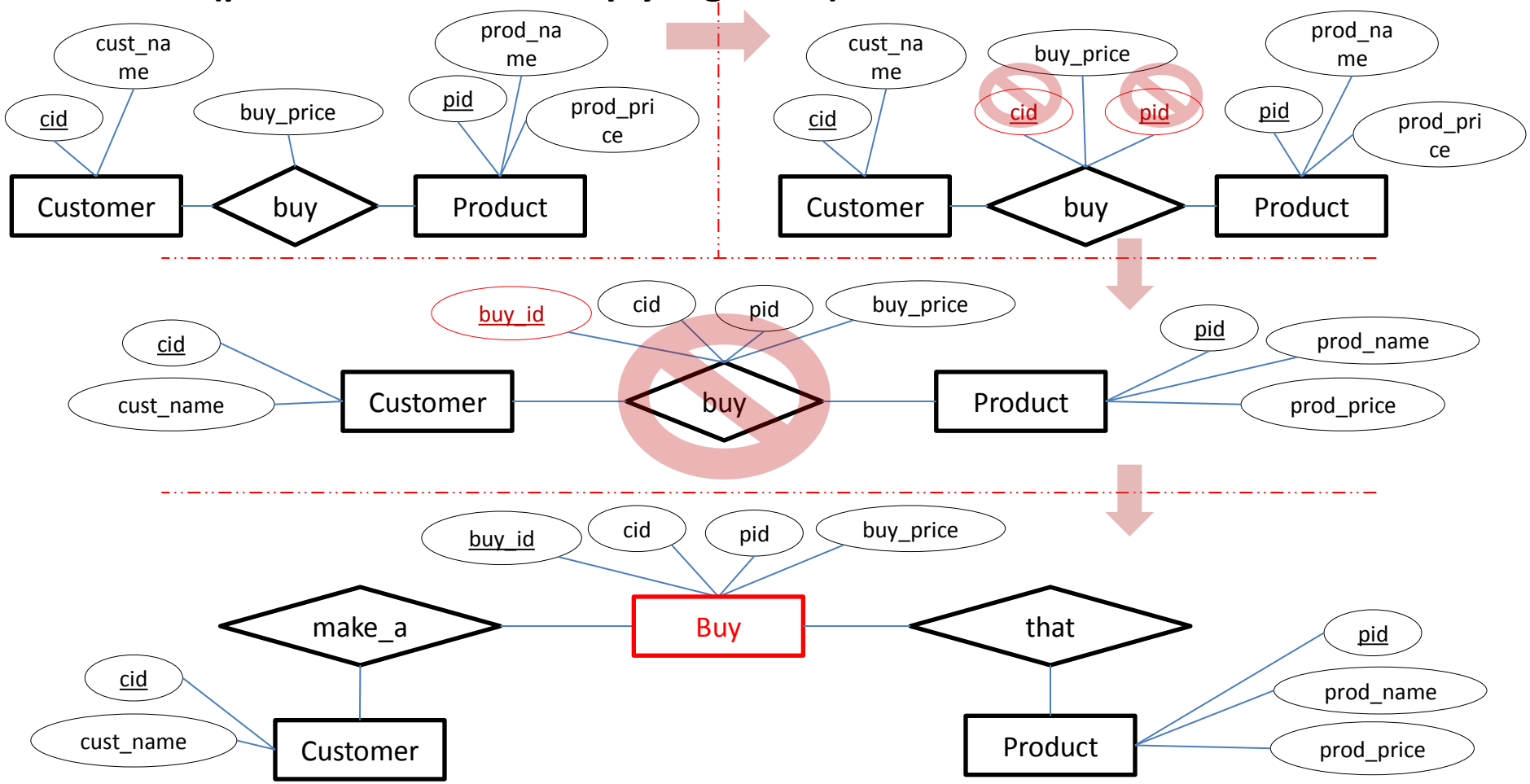
Contoh (perbaikan relationship yang salah):



- Meskipun entity order tidak berkaitan langsung ke relationship, namun jelas bahwa terdapat sejumlah **relationship** yang mungkin bisa kita tentukan dalam hal entity **order** dengan entity **Customer**, **Agent**, dan **Product**.

Intro ER Concept (Relationships Among Entities)

Contoh (perbaikan relationship yang salah):



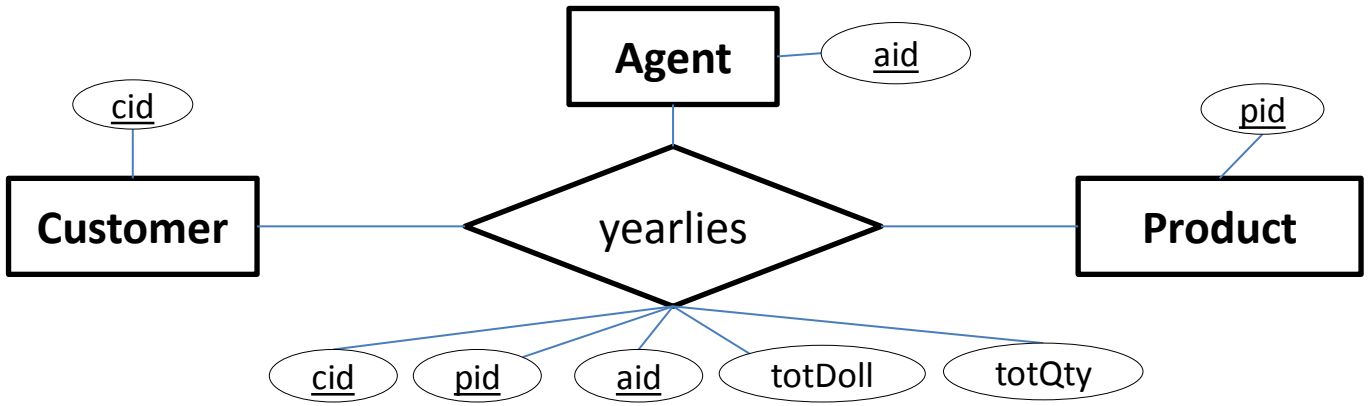
ER Concept (Relationships Among Entities)

Contoh (ternary relationship):

- Misal akan dibuat tabel yearlies sebagai berikut:

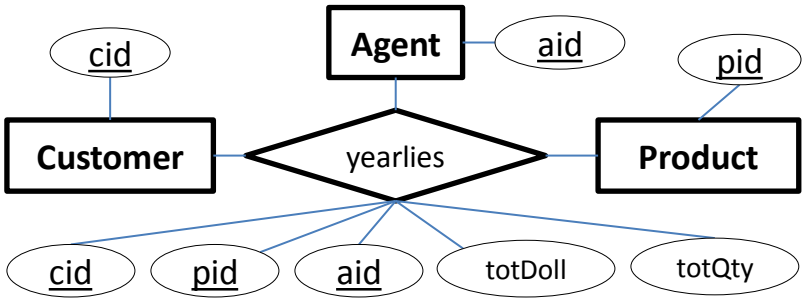
```
create table yearlies (cid char(4), aid char(3), pid char(3), totqty integer, totdoll float);
insert into yearlies
        select cid, aid, pid, sum(qty), sum(dollars) from orders group by cid, aid, pid;
```

- Berdasarkan sql di atas maka yearlies merupakan relationship terhadap Customer, Agent, Product.
- Relationship yearlies dikatakan **ternary relationship**, karena merupakan relasi terhadap 3 entity.



ER Concept (Relationships Among Entities)

Contoh (ternary relationship):



- ***n-ary* Relationship** (dengan *n > 2*) biasanya dipecah menjadi sejumlah **binary relationship** yang berbeda-beda.
- Dalam beberapa kasus, *n-ary* relationship tidak dapat dipecah menjadi binary relationship (contohnya: yearlies).

ER Concept (Kesimpulan)

- **Entities dan Attributes**

Classification	Description	Example
Entity	A collection of distinguishable real-world objects with common properties	Customers, Agents, Products, Employees
Attribute	A data item that describes a property of an entity or relationship	cid, cust_name, pid, prod_name, prode_price
Identifier (set of attributes)	Uniquely identifies an entity or relationship occurrence	customer identifier: cid, employee identifier: eid
Descriptor	Non-key attribute, describing an entity or relationship	prod_price (for Product), Percent (for work_on), totDoll, totQty (for yearlies)
Composite attribute	A group of simple attributes that together describe a property of an object	student_name
Multi-valued attribute	An entity attribute that takes on multiple values for a single entity instance	hobbies

ER Concept (Kesimpulan)

- **Relationships**

Classification	Description	Example
Relationship	Named set of m-tuples, identifies subset of the Cartesian product E1 × E2 × . . . × Em	
Binary relationship	A relationship on two distinct entities	**Teaches** (for Instructor and Course_section), **works_on** (for Employee and Project)
Ring, recursive relationship	A relationship relating an entity to itself	**manages** (for Employee)
Ternary relationship	A relationship on three distinct entities	**yearlies** (for Customer, Agent dan Product)

ER Concept (Kesimpulan)

- Hal yang harus diperhatikan dalam **membuat diagram ER**:
  - **Atribut multivalue** dirubah menjadi entity dengan menyertakan atribut identifier entity asal dan atribut multivalue tersebut sebagai atribut pada entity tersebut.
  - **Relationship yang memiliki atribut** harus ditambahkan atribut identifier entity-entity yang terelasi, sebagai penghubung relationship dengan entity-entity tersebut dan atribut identifier entity-entity tersebut menjadi identifier kombinasi pada relationship.
  - **Relationship yang identifiernya bukan kombinasi identifier** dari identifier entity-entity (yang terelasi) tidak bisa dijadikan relationship, harus dirubah menjadi Entity dan dibuatkan relationship yang menghubungkan entity baru tersebut dengan entity-entity yang sebelumnya terelasi.

ER Concept (Kesimpulan)

- Hal yang harus diperhatikan pada **transformasi ER menjadi tabel**:
  - Semua entity ditansformasi menjadi tabel.
  - Kolom pada tabel dibuat dari atribut dan subset atribut komposit, kecuali;
    - atribut multivalue (karena akan menjadi entity).
    - atribut komposit (hanya subset-nya yang menjadi kolom).
  - Relationship yang memiliki atribut ditransformasi menjadi tabel.
    - Relationship yang identifier-nya bukan kombinasi identifier dari identifier entity-entity yang terelasi, harus dirubah menjadi entity dan dibuatkan beberapa relationship baru yang menghubungkannya dengan entity-entity yang sebelumnya terelasi.
    - Relationship yang tidak memiliki atribut tidak ditransformasi menjadi tabel, namun atribut identifier entity yang dirujuk dijadikan atribut pada entity yang merujuk. Dengan demikian berarti menjadi kolom pada tabel hasil transformasi entity yang merujuk.

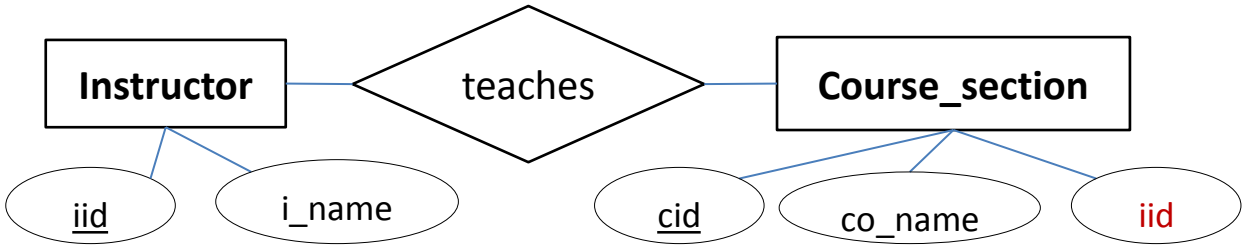
Cardinality

Partisipasi Entity dalam suatu Relationship

Ilustrasi untuk Memahami Cardinality

Perhatikan diagram ER dan rule berikut;

- Pada relationship teaches (Instructor, teaches, Course_section) memiliki aturan bahwa;
  - Setiap Course_section harus memiliki setidak-tidaknya satu instructor yang ditugaskan mengajar.
  - Setiap Course_section hanya boleh diajar oleh satu instructor (maksimum diajar oleh satu instructor).
  - Seorang instructor dimungkinkan untuk tidak mengajar suatu course.
  - Seorang instructor dimungkinkan untuk mengajar lebih dari satu course.

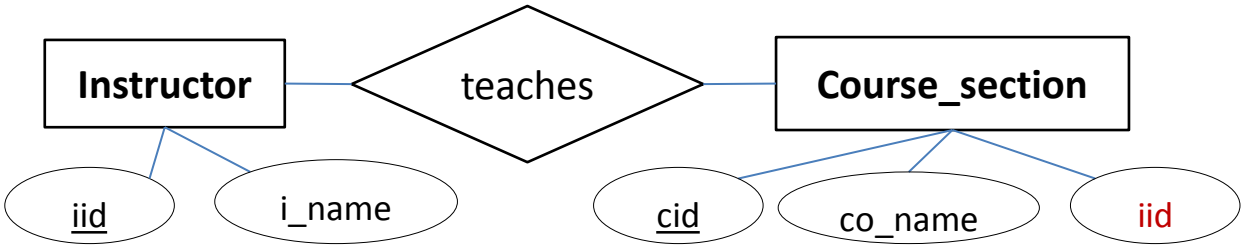


Bersambung ...

Cardinality

Partisipasi Entity dalam suatu Relationship

Ilustrasi untuk Memahami Cardinality



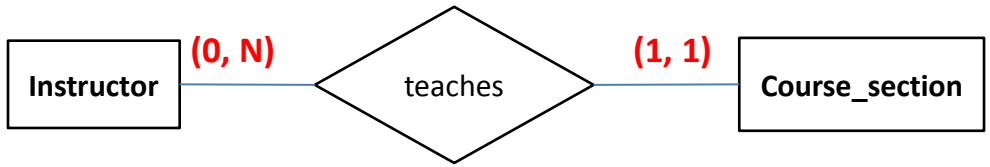
instructor

iid	i_name
001	Budi
002	Ani
003	Dedi

teaches

course_section

cid	co_name	iid
C01	Database 1	001
C02	Database 2	001
C03	Analysis	002



instructor **teaches** **course_section**

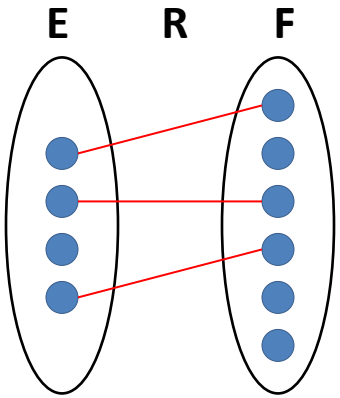
iid		iid
001		001
002		001
003		002

min-card(instructor, teaches) = 0
max-card(instructor, teaches) = n
min-card(course_section, teaches) = 1
max-card(course_section, teaches) = 1
Many-to-one Relationship, dengan course_section pada posisi many

Cardinality

Partisipasi Entity dalam suatu Relationship

E dan F adalah entity, dan R adalah relationship



One-to-one relationship

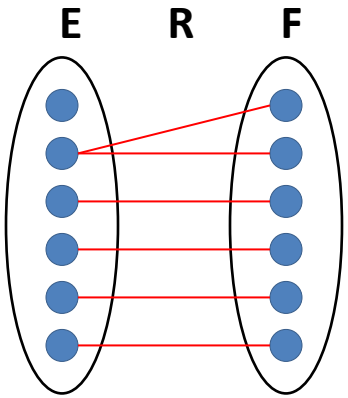
min-card(E, R) = 0

max-card(E, R) = 1

min-card(F, R) = 0

max-card(F, R) = 1

(a)



Many-to-one relationship

min-card(E, R) = 0

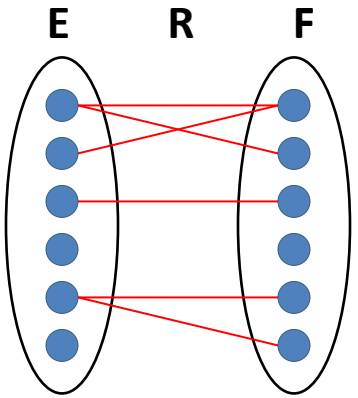
max-card(E, R) = N

min-card(F, R) = 1

max-card(F, R) = 1

F is the "many" side here

(b)



Many-to-many relationship

min-card(E, R) = 0

max-card(E, R) = N

min-card(F, R) = 0

max-card(F, R) = N

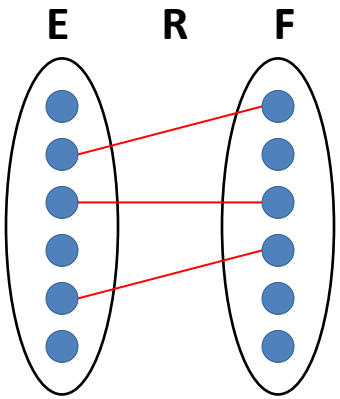
(c)

Bersambung …

Cardinality

Partisipasi Entity dalam suatu Relationship

E dan F adalah entity, dan R adalah relationship



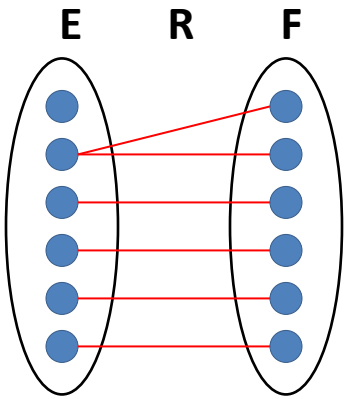
One-to-one relationship

min-card(E, R) = 0

max-card(E, R) = 1

min-card(F, R) = 0

max-card(F, R) = 1



Many-to-one relationship

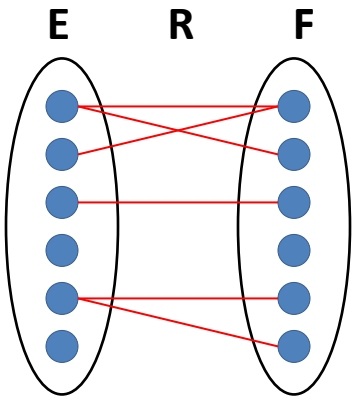
min-card(E, R) = 0

max-card(E, R) = N

min-card(F, R) = 1

max-card(F, R) = 1

F is the "many" side here



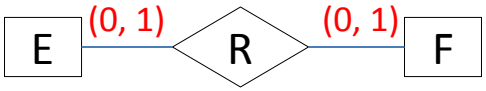
Many-to-many relationship

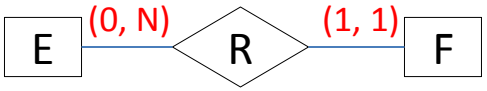
min-card(E, R) = 0

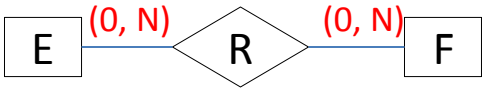
max-card(E, R) = N

min-card(F, R) = 0

max-card(F, R) = N

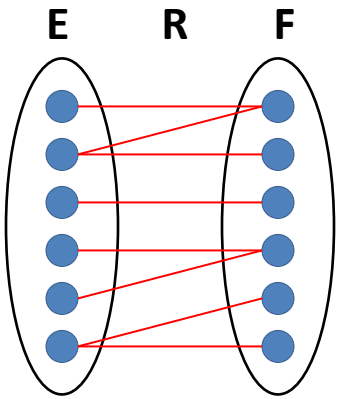


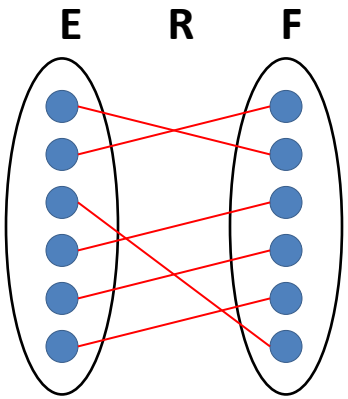


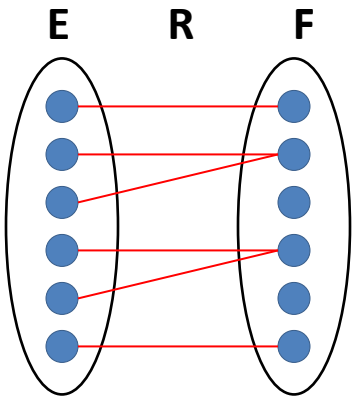


Quiz

Tentukan cardinality







?

min-card(E, R) = ?

max-card(E, R) = ?

min-card(F, R) = ?

max-card(F, R) = ?

?

min-card(E, R) = ?

max-card(E, R) = ?

min-card(F, R) = ?

max-card(F, R) = ?

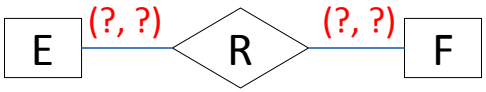
?

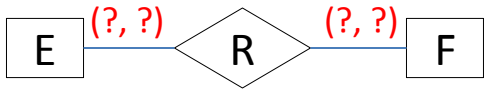
min-card(E, R) = ?

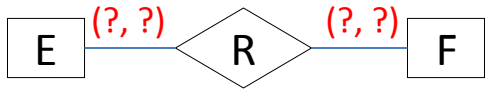
max-card(E, R) = ?

min-card(F, R) = ?

max-card(F, R) = ?







See You Next Session

- **Thank's**