# Entity-Relationship modeling

# TIK

- The basic concepts associated with the Entity–Relationship (ER) model, namely entities, relationships, and attributes.
- How to use Entity–Relationship (ER) modeling in database design.

# Outline

- Entity Relationship Model
- Entity/Relationship Diagrams
- E-R Notation
- Entities
- Attributes
- Relationship

# Outline

- Cardinalities
- Strong and Weak Entity
- Problems in E-R Model

# Entity Relationship Model

- Model E-R merupakan salah satu dari model yang berdasarkan objek
- Teknik ini digunakan dalam fase desain konseptual
- Pertama kali diperkenalkan oleh Peter Chen pada tahun 1976
- Sekarang notasi dan variasi model E-R telah banyak dan berkembang

# Entity Relationship Model

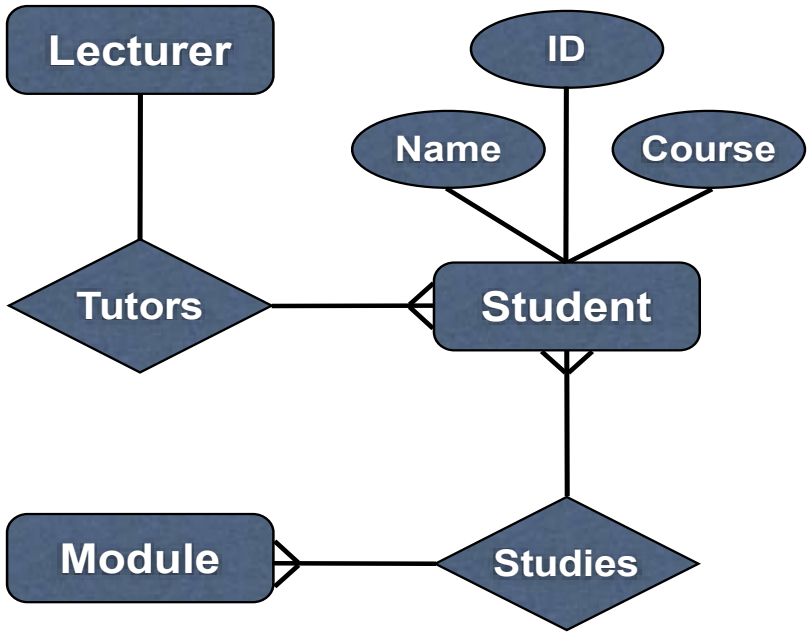
- Tujuan Model
  - Melihat dan memahami masalah pemodelan secara menyeluruh
  - Mengenal secara pasti proses dan konstrain
  - Membantu dalam implementasi  database
  - Membantu menghasilkan model yang stabil dan tidak berulang

# Entity Relationship Models

- Suatu model yang digunakan untuk menggambarkan data dalam bentuk
  - Entities:
    - Objects or items of interest
  - Attributes:
    - Facts about or properties of an entity
  - Relationships:
    - Links between entities

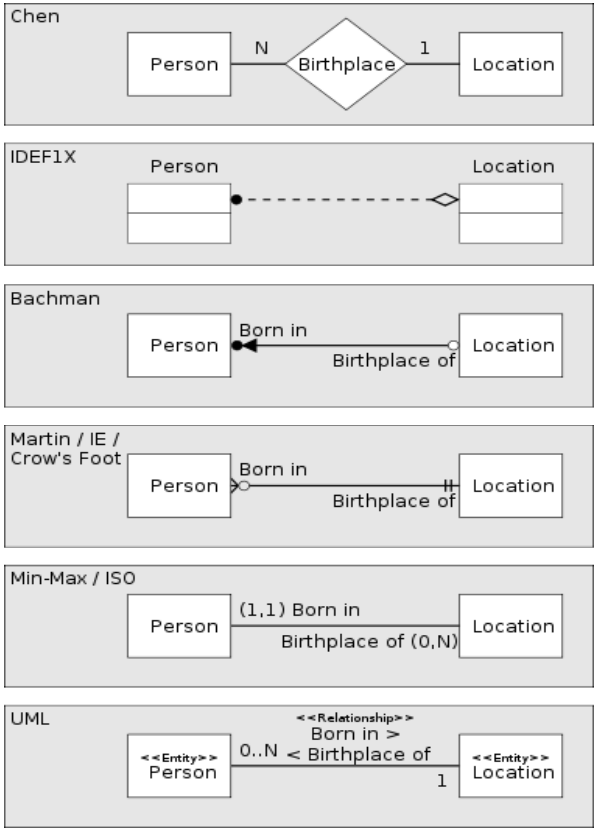
# Entity/Relationship Diagrams

- E/R Models are often represented as E/R diagrams that
  - Give a conceptual view of the database
  - Are also independent of the choice of DBMS
  - Can identify some problems in a design



# E-R Notation

- Various conventions
- Chen
  - Mostly American
  - Manga guide
- Crow's Foot
  - Mostly British
  - Many others



# Entities

- A *database* can be modeled as:
  - a collection of entities,
  - relationship among entities.

- **Entitas** adalah sesuatu atau objek di dunia nyata yang dapat dibedakan dengan sesuatu atau objek lainnya.
  - sesuatu yang **nyata** (seseorang, tempat, objek, kejadian)
  - **abstrak** (berupa suatu konsep)

# Entities

- Contoh entitas:
  - *Orang* : Pegawai, Mahasiswa, Pasien
  - *Tempat* : Toko, Gedung, Propinsi
  - *Objek* : Mesin, Gedung, Mobil
  - *Kejadian* : Penjualan, Registrasi
  - *Konsep*: Rekening, Kursus, Gaji

# Entities

- Entities have
  - A type or class, such as Fruit
  - Instances of that type, such as Apple or Lemon
  - Attributes (such as variety and price)

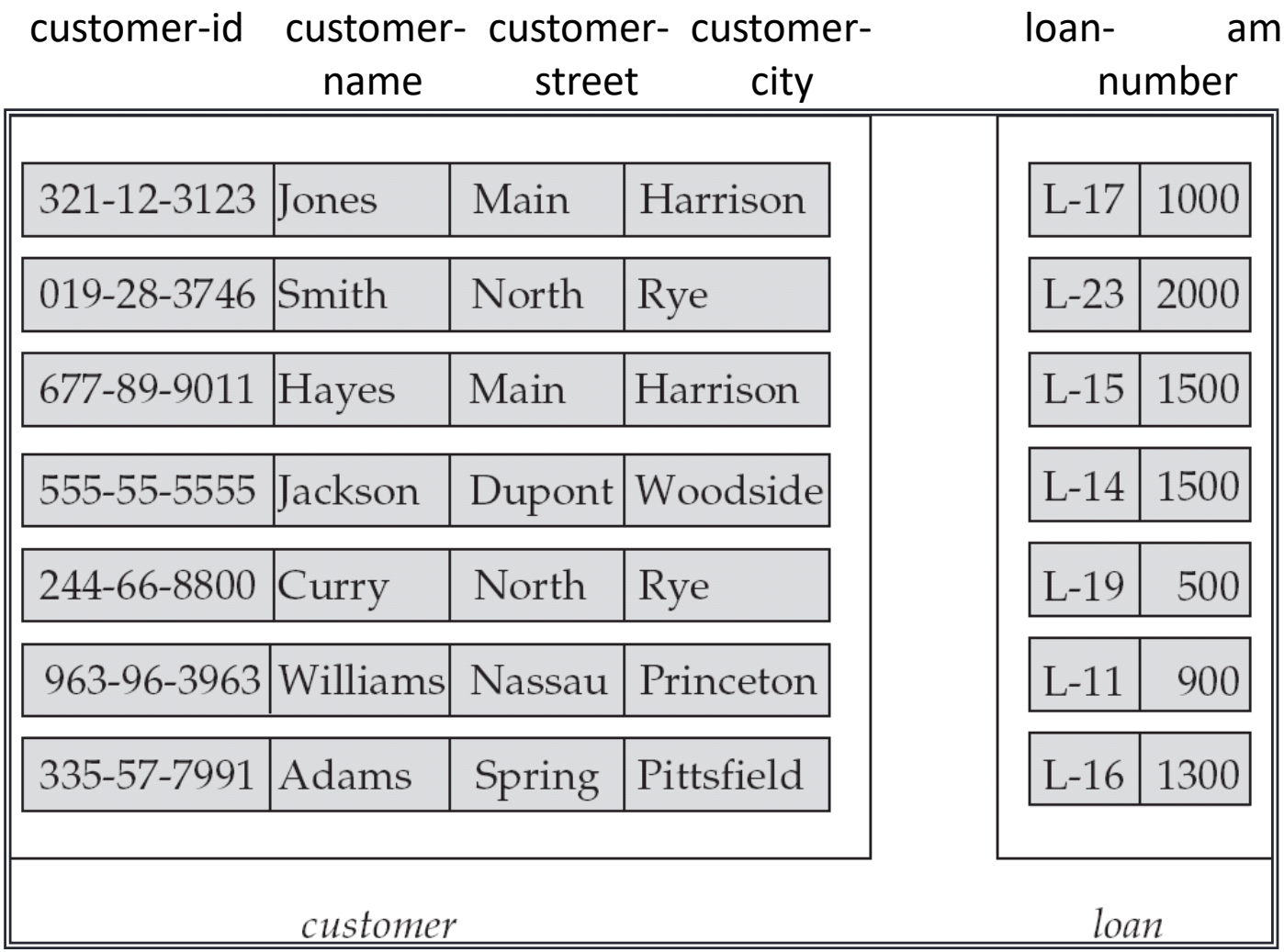
# Entities

| Atribut | Tipe Data Atribut | Instance 1 | Instance 2 |
|---|---|---|---|
| Nomor_Mahasiswa | CHAR(5) | 08001 | 08002 |
| Nama | CHAR(35) | Emi | Sutanto |
| Tanggal_Lahir | Date | 20/08/1990 | 04/05/1991 |
| Jenis_Kelamin | CHAR(6) | Wanita | Pria |

Tipe entitas Mahasiswa dengan dua buah Instance

# Diagramming Entities

- Sebuah entitas dinyatakan dengan **kata benda tunggal**
- digambarkan dengan bentuk **kotak (rectangle)**


customer-id
customer-name
customer-street
customer-city
321-12-3123
Jones
Main
Harrison
019-28-3746
Smith
North
Rye
677-89-9011
Hayes
Main
Harrison
555-55-5555
Jackson
Dupont
Woodside
244-66-8800
Curry
North
Rye
963-96-3963
Williams
Nassau
Princeton
335-57-7991
Adams
Spring
Pittsfield
customer
loan-number
amount
L-17
1000
L-23
2000
L-15
1500
L-14
1500
L-19
500
L-11
900
L-16
1300
loan

# Diagramming Entities


Penggambaran entity set dalam model E-R

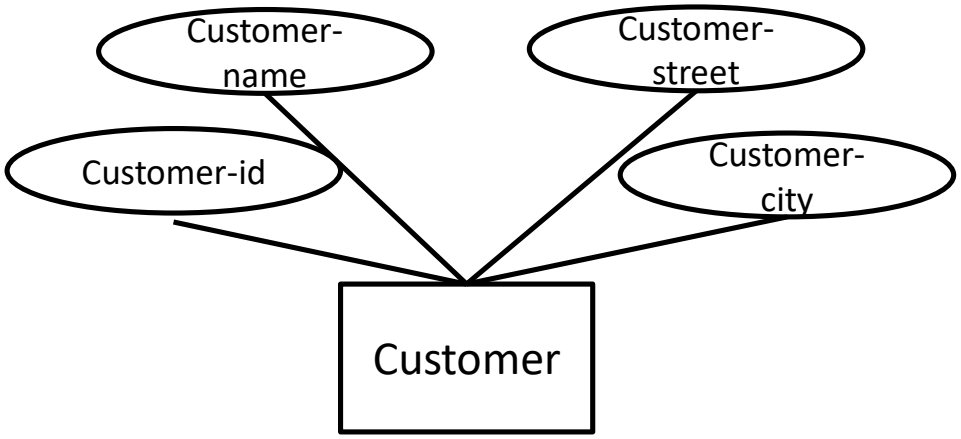
# Attributes

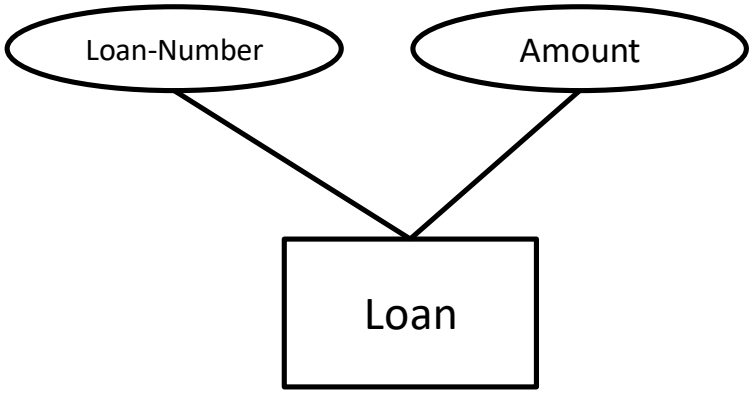
- Attributes are facts, aspects, properties, or details about an entity
  - customer have id, name, street, city,…
  - loan have number, amount,…
- Attributes have
  - A name
  - An associated entity
  - Domains of possible values
  - Values from the domain for each instance of the entity they are belong to

# Diagramming Attributes

- In an E/R Diagram attributes may be drawn as ovals
- Each attribute is linked to its entity by a line
- The name of the attribute is written in the oval

# Diagramming Attributes



# Attributes

- Attributes can be classified as:
  - **Simple** and **composite** attributes.
  - **Single-valued** and **multi-valued** attributes.
  - **Derived** attributes

# Simple and Composite Attributes

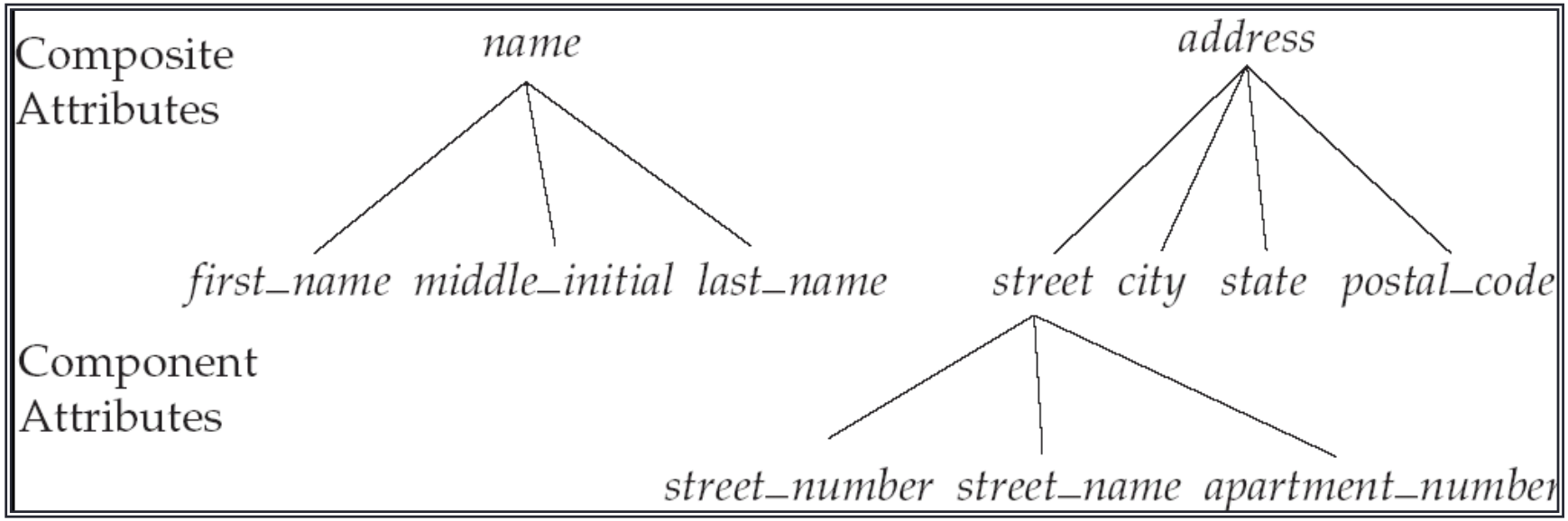
- **Simple (sedehana)**
  - Atribut yang tidak dapat dipecah menjadi bagian-bagian yang lebih kecil yang masih memiliki makna
    - Contoh,

      `Jenis_Kelamin (Wanita dan Pria)`

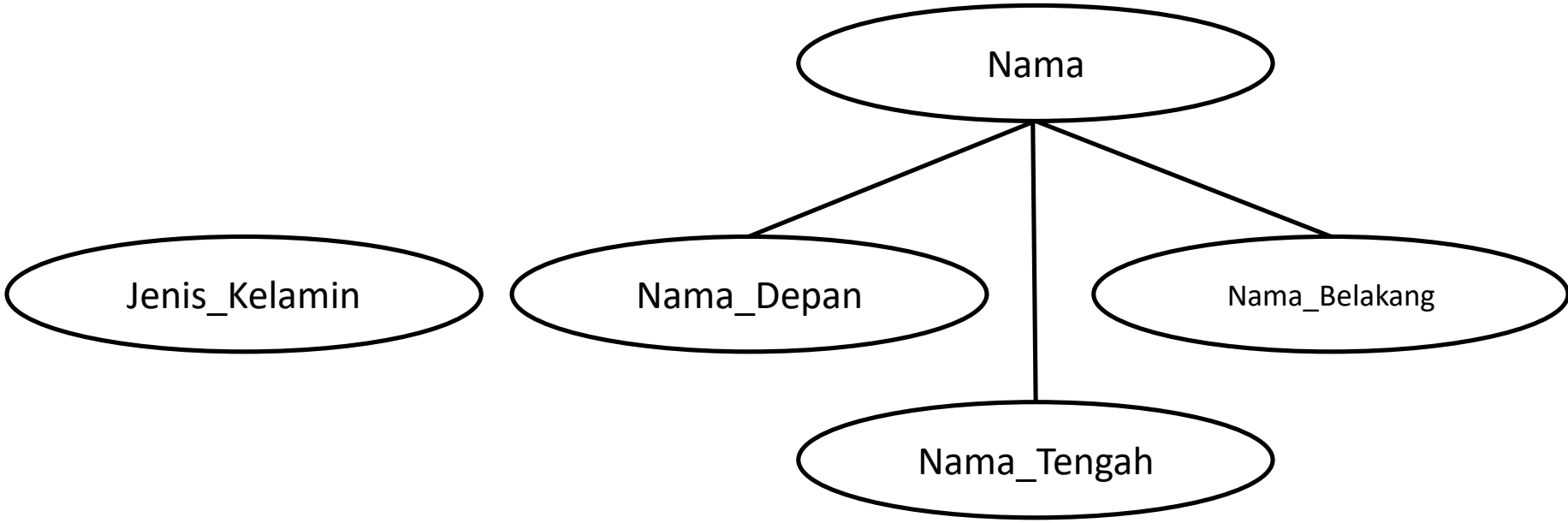
# Simple and Composite Attributes

- **Composite Attributes**
  - Suatu atribut dapat dipecah menjadi bagian-bagian yang lebih kecil dan tetap bermakna.
    - Contoh,
      - `Nama : Nama_Depan, Nama_Tengah, Nama_Belakang`
      - `Alamat : Jalan, Kota, Kode_Pos`
      - `Jalan : Nomor_Jalan, Nama_Jalan, Nomor_Apartemen`

# Simple and Composite Attributes



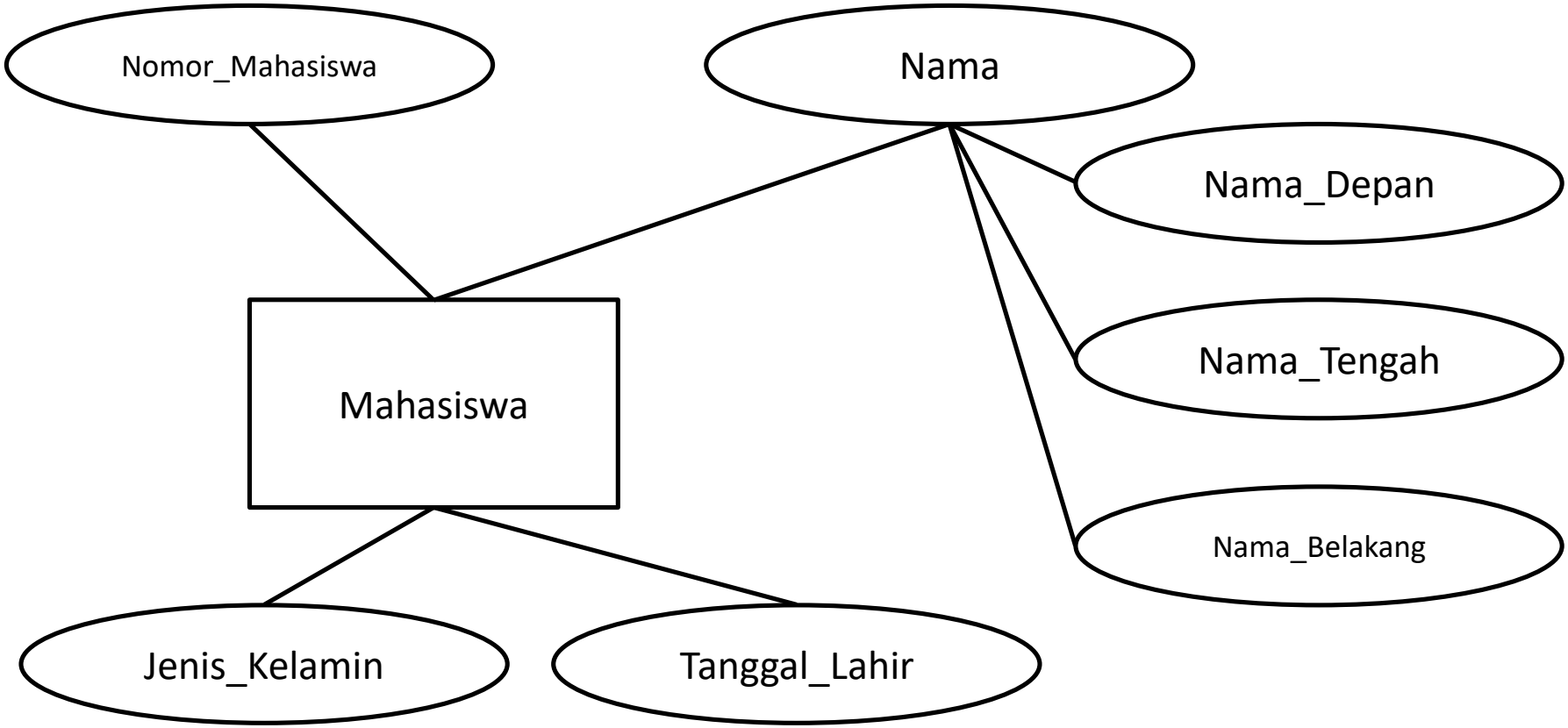
# Diagramming Simple and Composite Attributes




(a) Atribut sederhana

(b) Atribut komposit

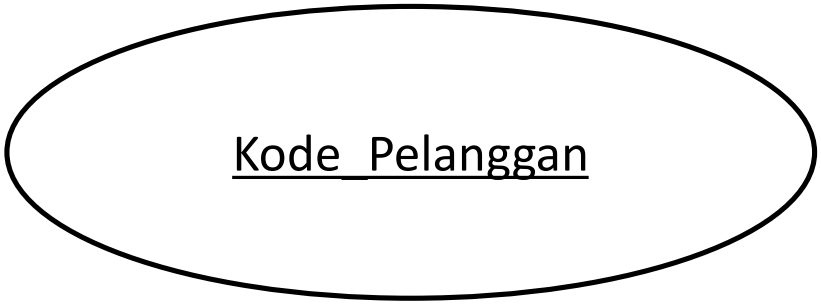
# Diagramming Simple and Composite Attributes




Entity yang melibatkan atribut kompisit dan sederhana

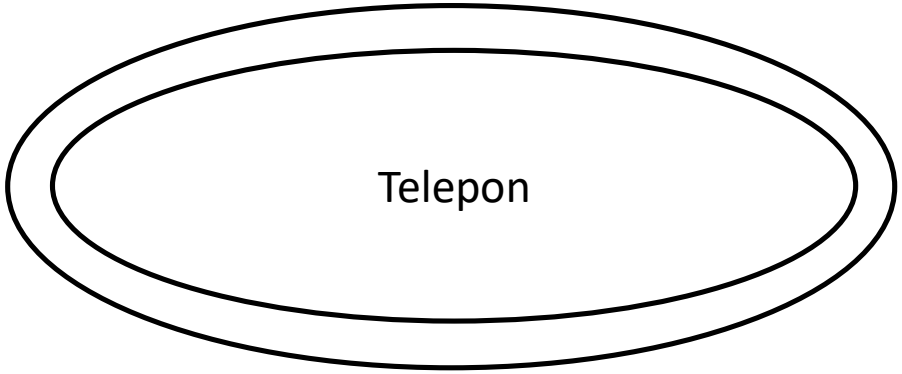
# Single-valued attributes

- **Atribut bernilai-tunggal (single-valued attribute)**
  - Atribut yang nilai atributnya hanya satu untuk setiap instance entity



# Multi-valued attributes

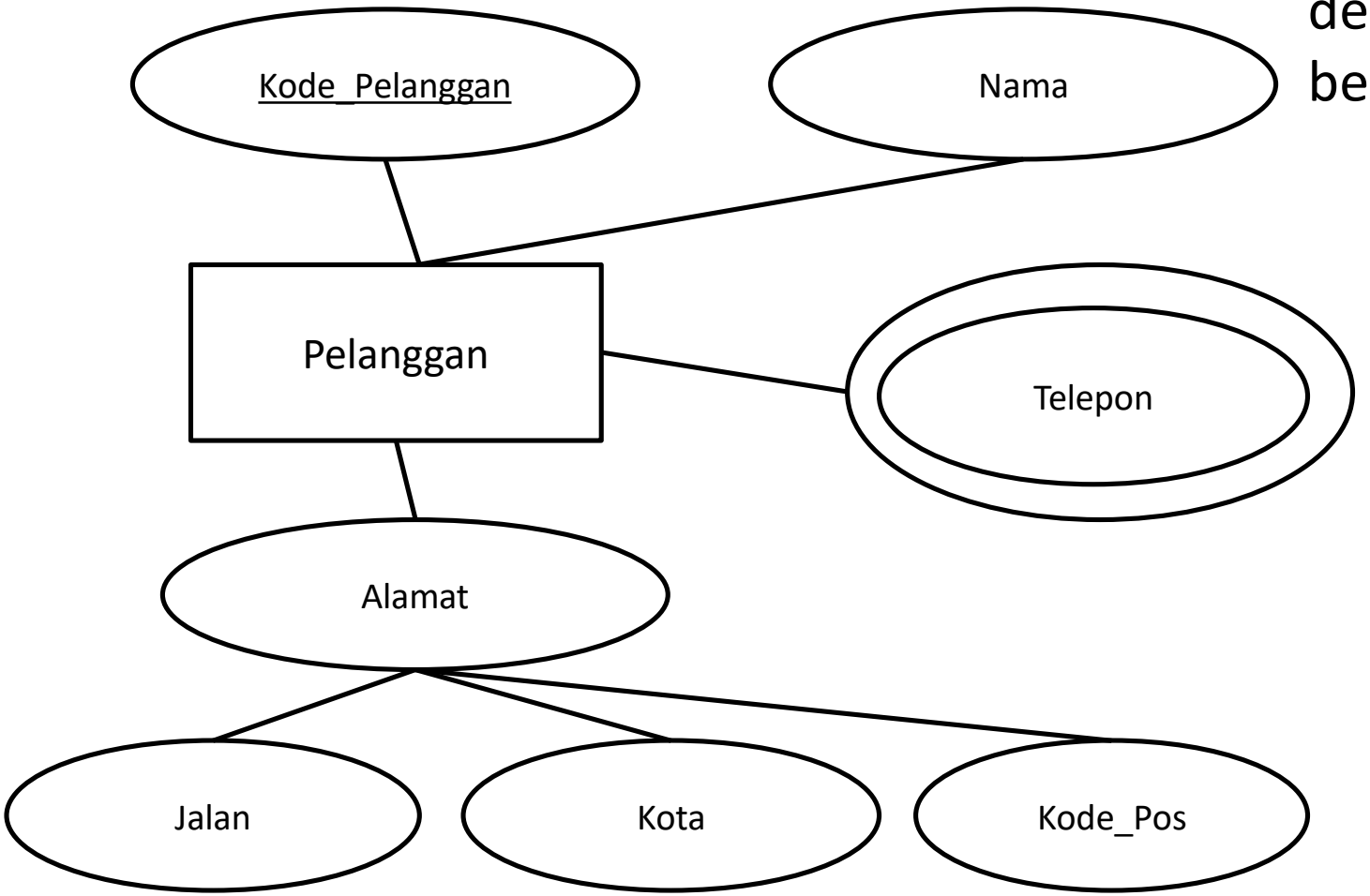
- Atribut bernilai-banyak (multi-valued attribute)
  - Atribut yang nilai atributnya bisa lebih dari satu untuk setiap instance entity
  - Contoh, Hobby, Telepon




Multi-valued attribute

# Diagramming Single-valued and multi-valued attributes

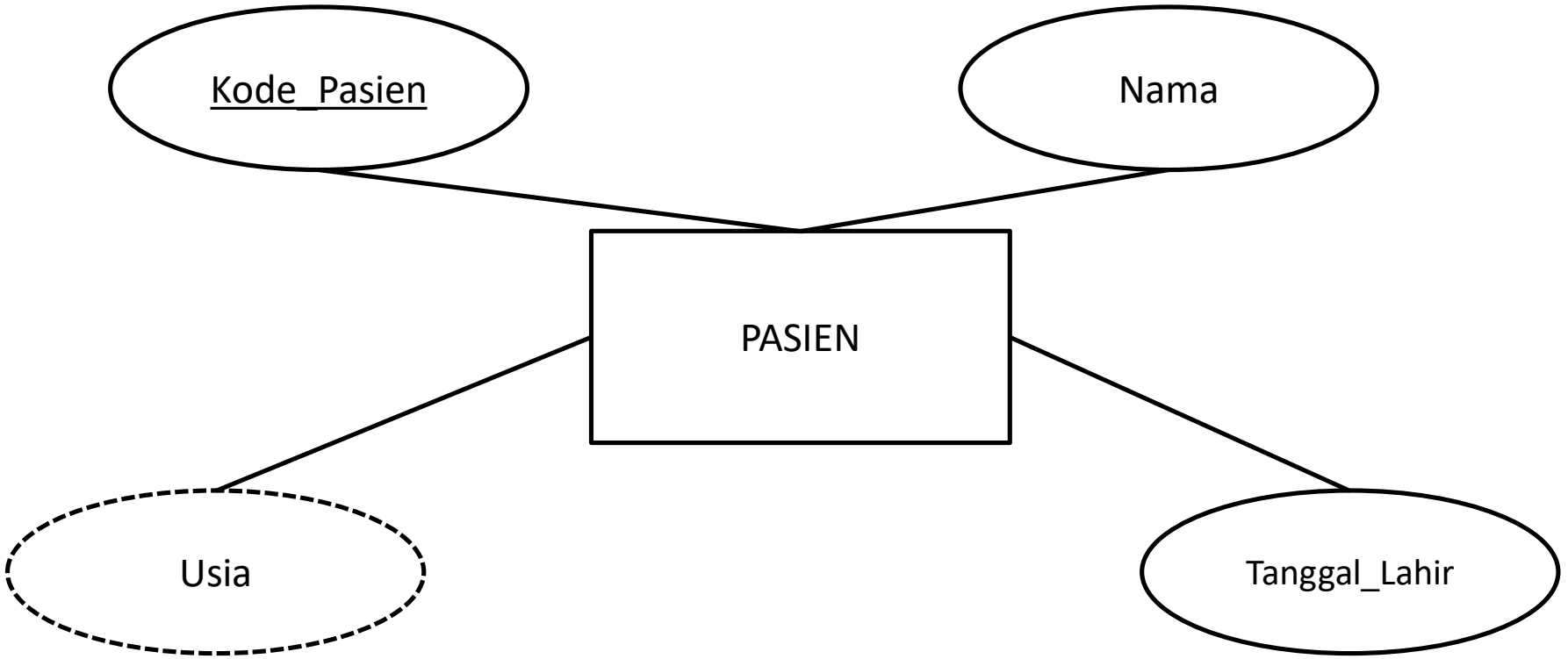
entity set dengan atribut bernilai banyak



# Derived Attribute

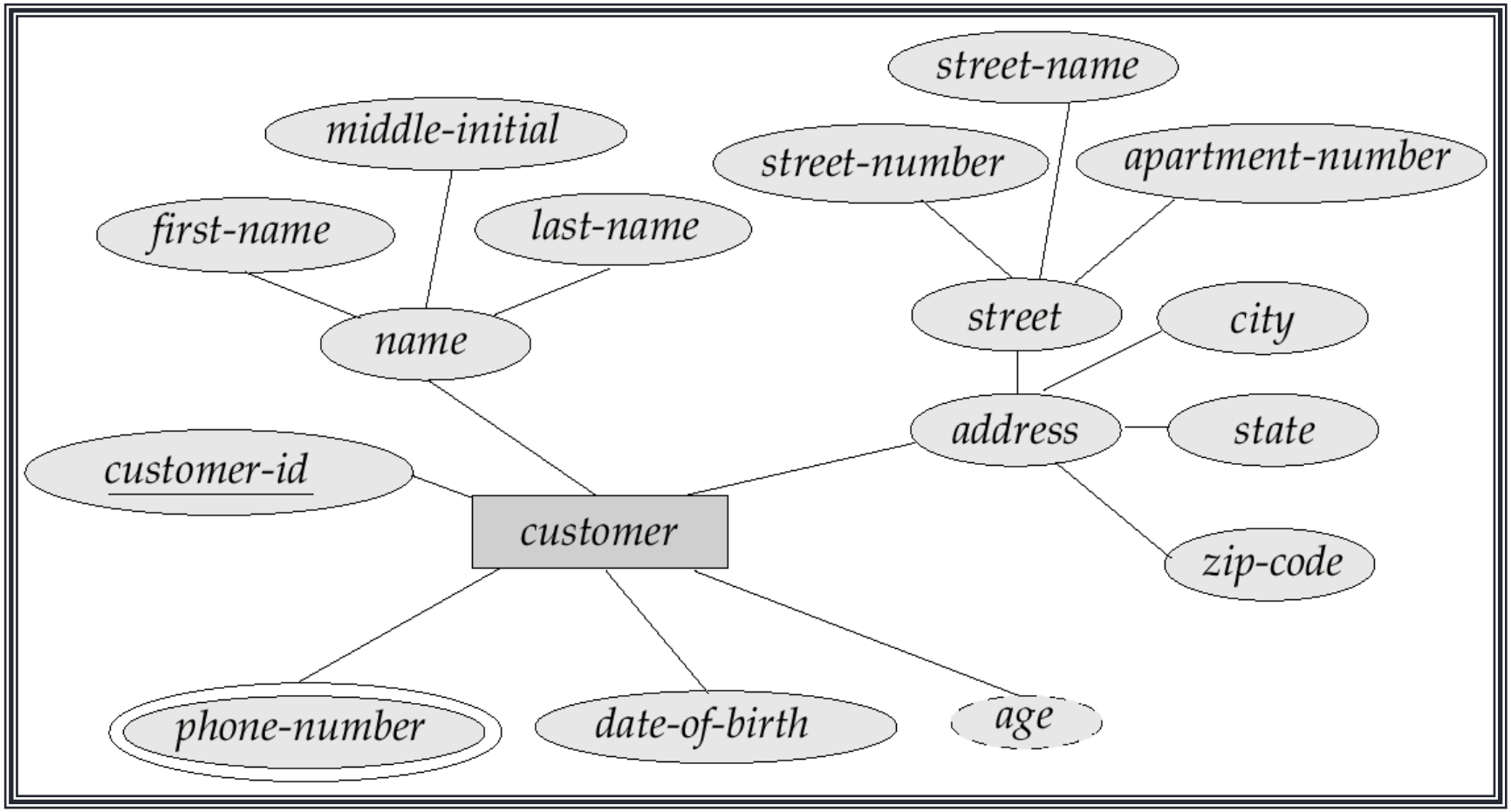
- Nilai atribut dalam satu entity set bisa saja dihitung atau diturunkan dari nilai suatu atribut atau sejumlah atribut yang tersimpan dalam database atau dari nilai lain (misal, jam sistem atau tanggal sistem)
- Contoh,
  - `Usia`, nilai atribut dapat diperoleh dari atribut `Tanggal_Lahir`
  - `Lama_Bekerja`, nilainya bisa dihitung berdasarkan `Tanggal_Mulai_Bekerja`.

# Diagramming Derived Attribute




entity set yang melibatkan atribut turunan yaitu usia

# E-R Diagram With Composite, Multivalued, and Derived Attributes



# Relationship

- Relationships are an association between two or more entities
- Relationships have
  - A name
  - A set of entities that participate in them
  - A degree - the number of entities that participate (most have degree 2)
  - A cardinality ratio

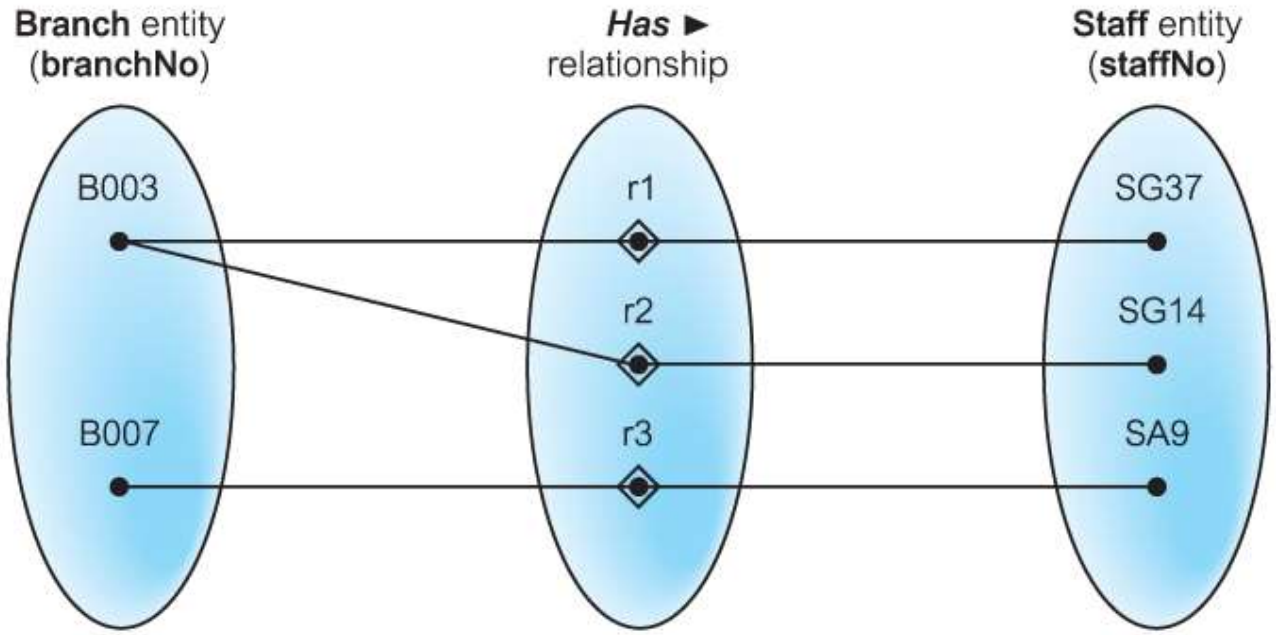
# Relationship

- Each relationship is shown as a **diamond** and **line** connecting the associated entity, labeled with the name of the relationship
- Normally, a relationship is named using a **verb**(`Supervises` or `Manages`)

# Relationship

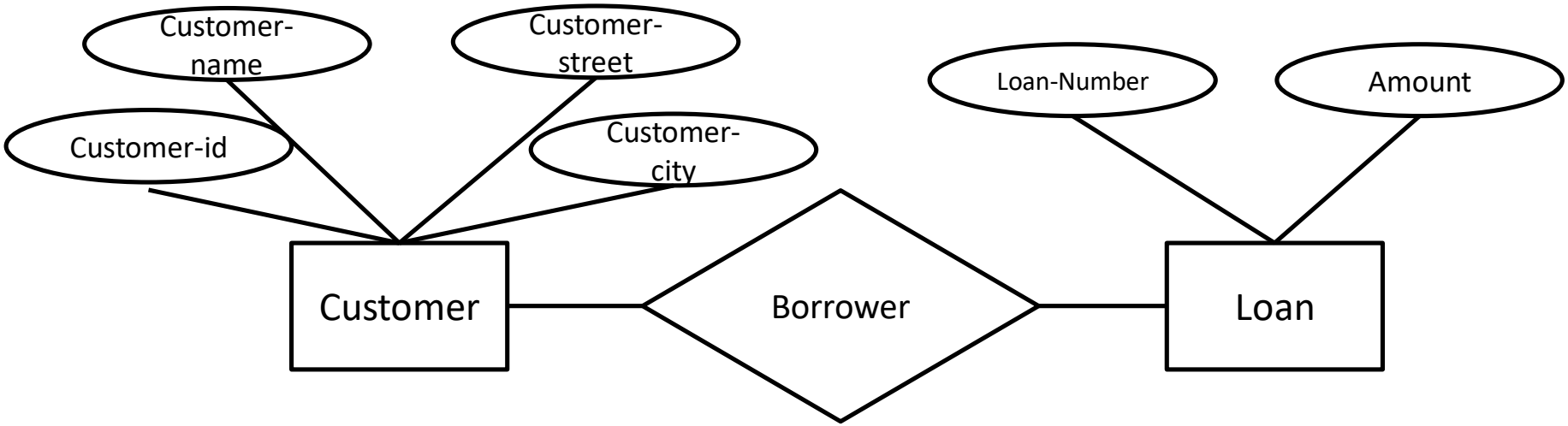
- **Relationship occurrence**
  - Uniquely identifiable association, which includes one occurrence from each participating entity
  - indicates the particular entity occurrences that are related

# Relationship




A semantic net showing individual occurrences of the Has relationship set.

# Diagramming Relationship



# Degree of Relationship

- Jumlah entity set yang dilibatkan oleh sebuah hubungan.
- Entity set yang dilibatkan dalam suatu hubungan disebut **partisipan**.
- The number of participants in a relationship set is called the **degree** of that relationship
- Therefore, the degree of a relationship indicates the number of entity sets involved in a relationship

# Degree of Relationship

- Relationship of degree :
  - **Binary**
  - **Tertiary**
  - **Quaternary**
  - **Unary/recursive**
- Generally, most relationship sets in a database system are binary or degree two

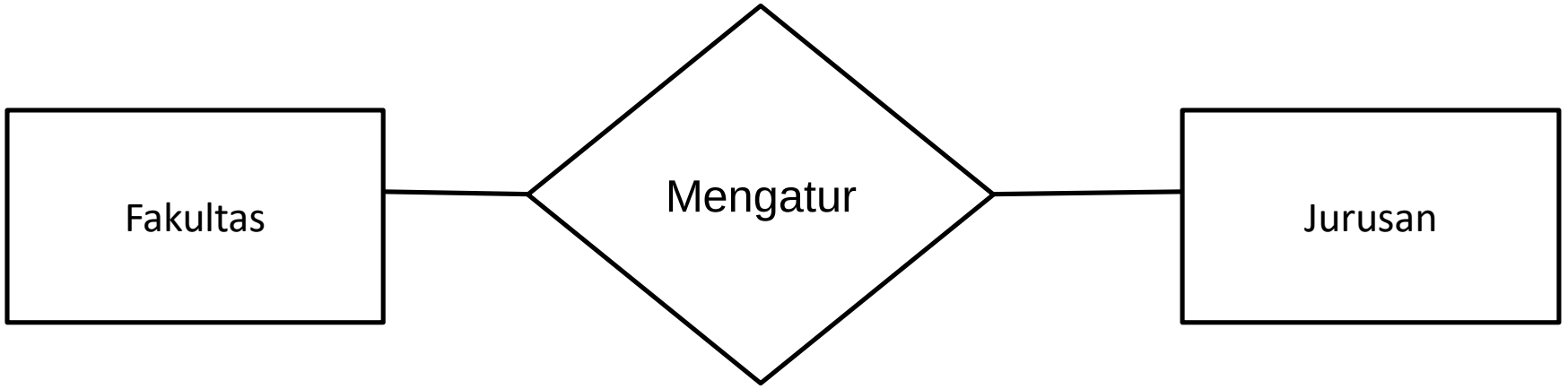
# Degree of Relationship

- Relationship sets may involve more than two entity sets.
  - E.g. Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets E*mployee, Job and Branch*

- Relationships between more than two entity sets are rare. Most relationships are binary.

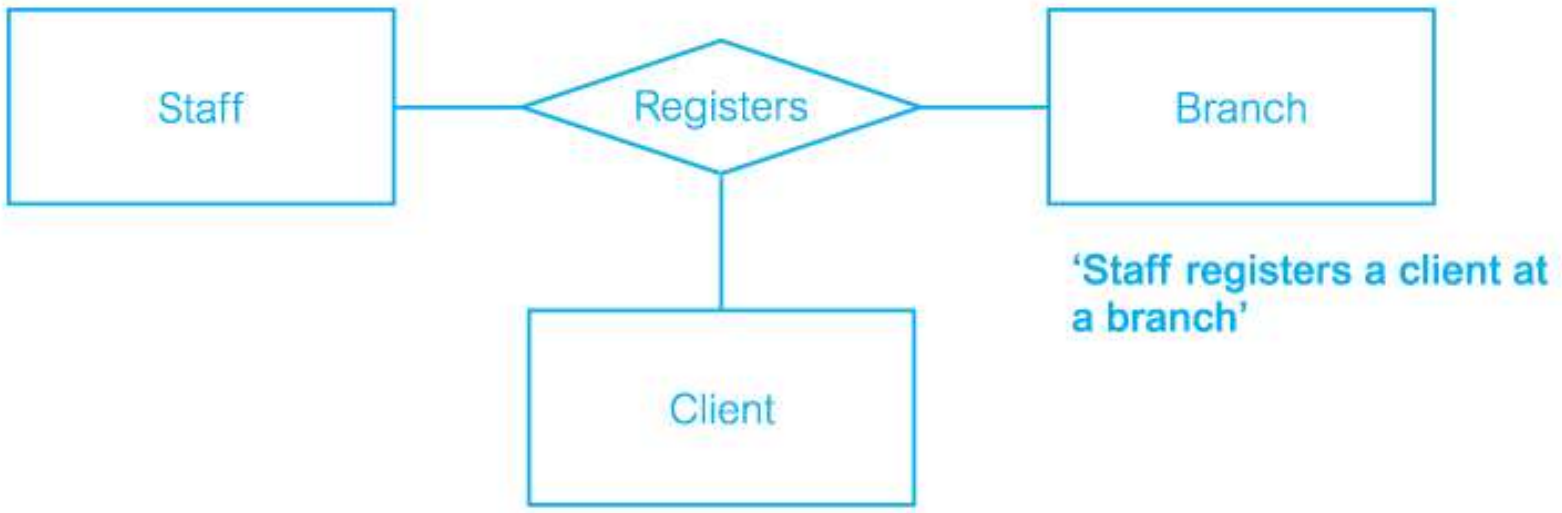
# Binary Relationship

- Hubungan yang melibatkan dua buah entity set.

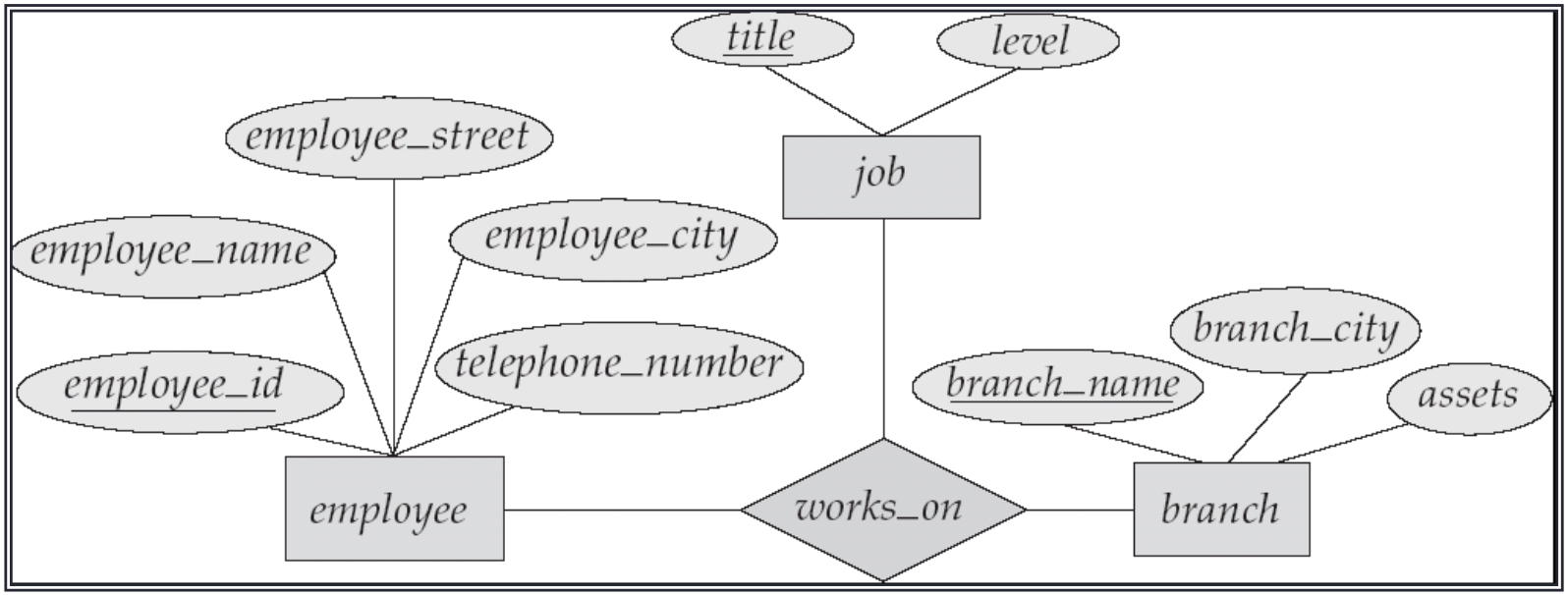


# Ternary Relationship

- Hubungan yang melibatkan tiga buah entity set.



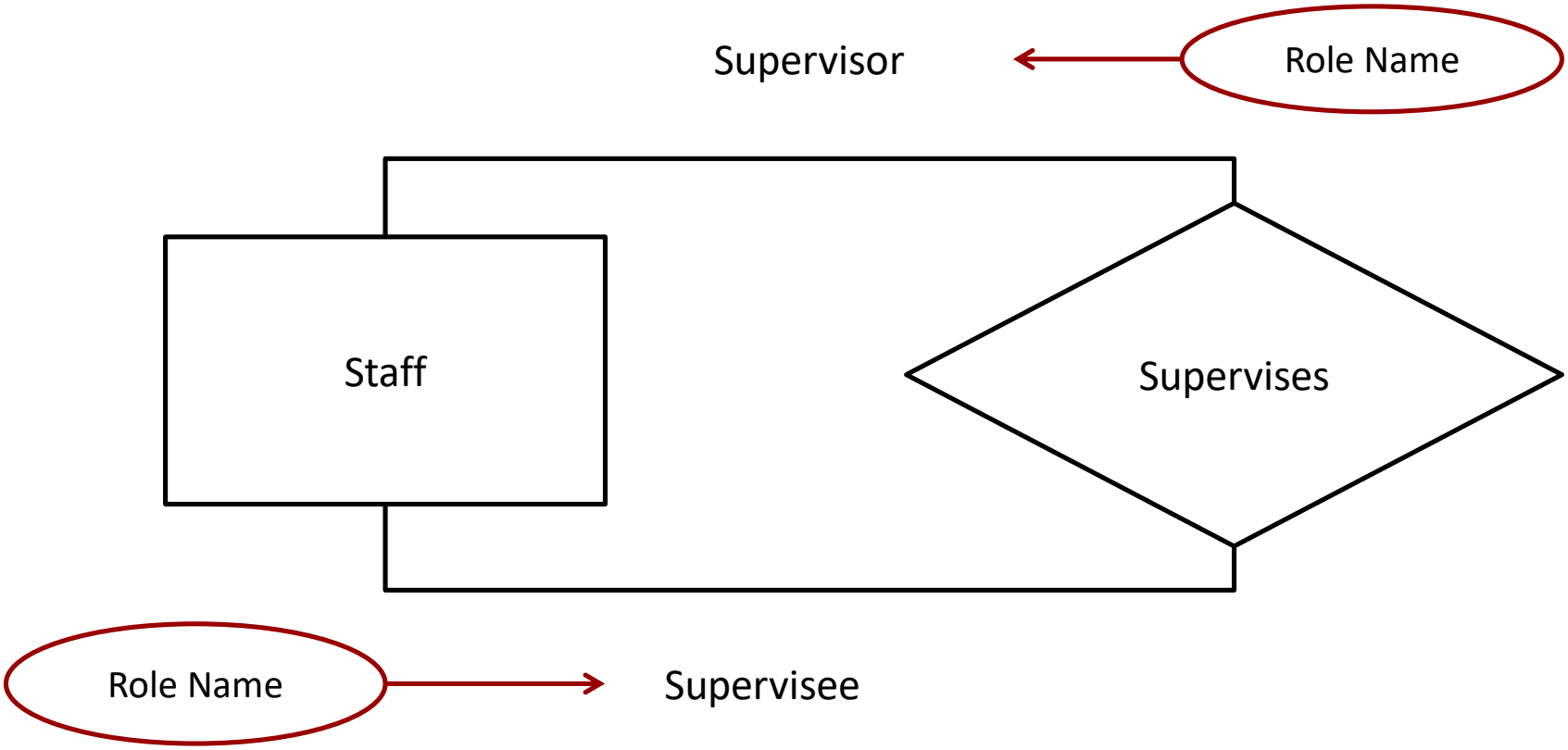
# Ternary Relationship



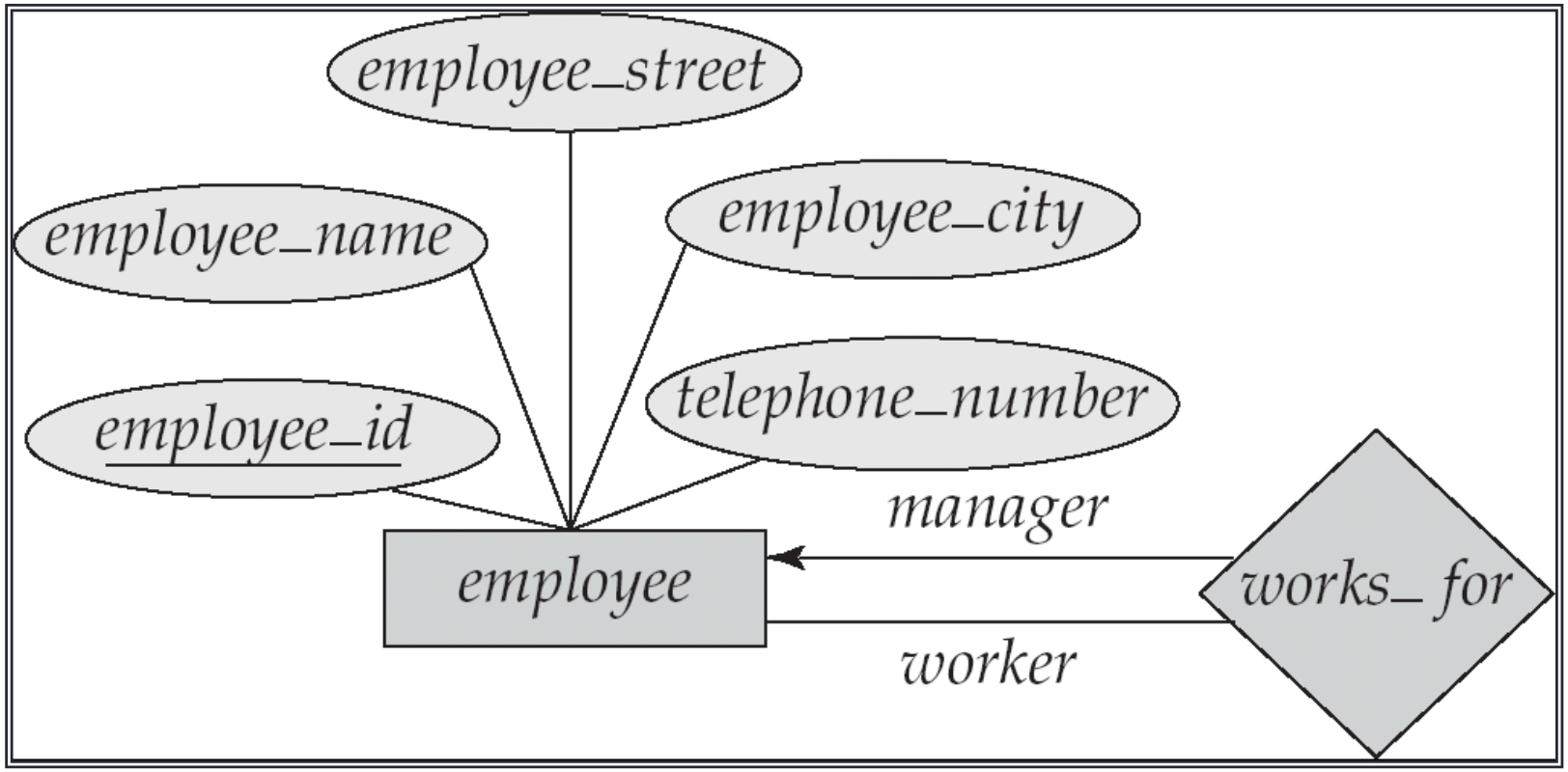
# Recursive (Unary) Relationship

- Hubungan yang melibatkan hanya satu entity set
- Relationship set where same entity set participates more than once in different roles.
- Relationships may be given **role names (nama peran)** to indicate the purpose that each participating entity set plays in a relationship.
- Role names can be important for recursive relationships to determine the function of each participant.

# Recursive (Unary) Relationship

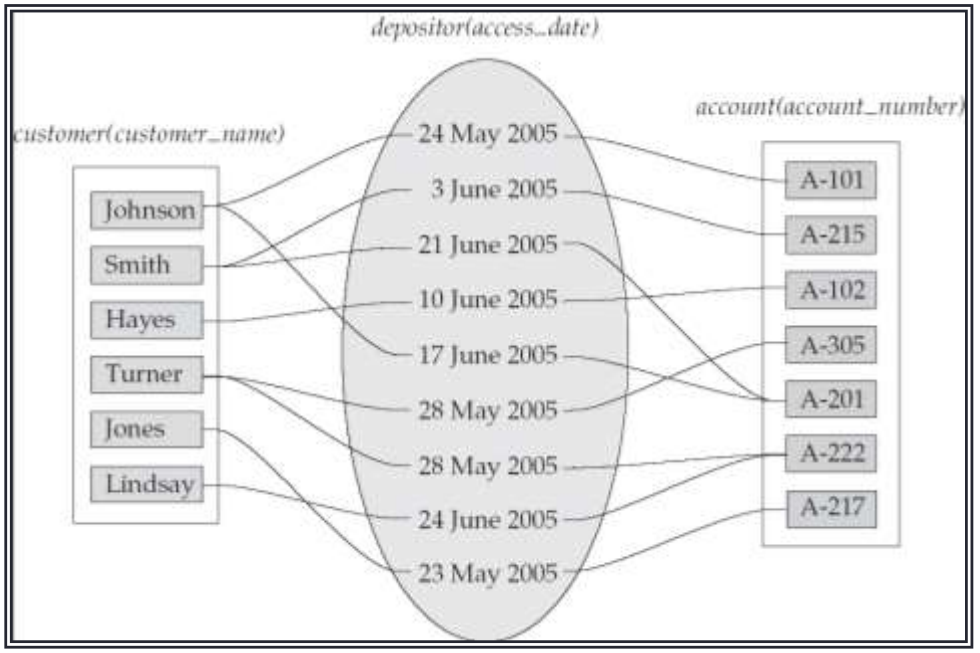


# Recursive (Unary) Relationship

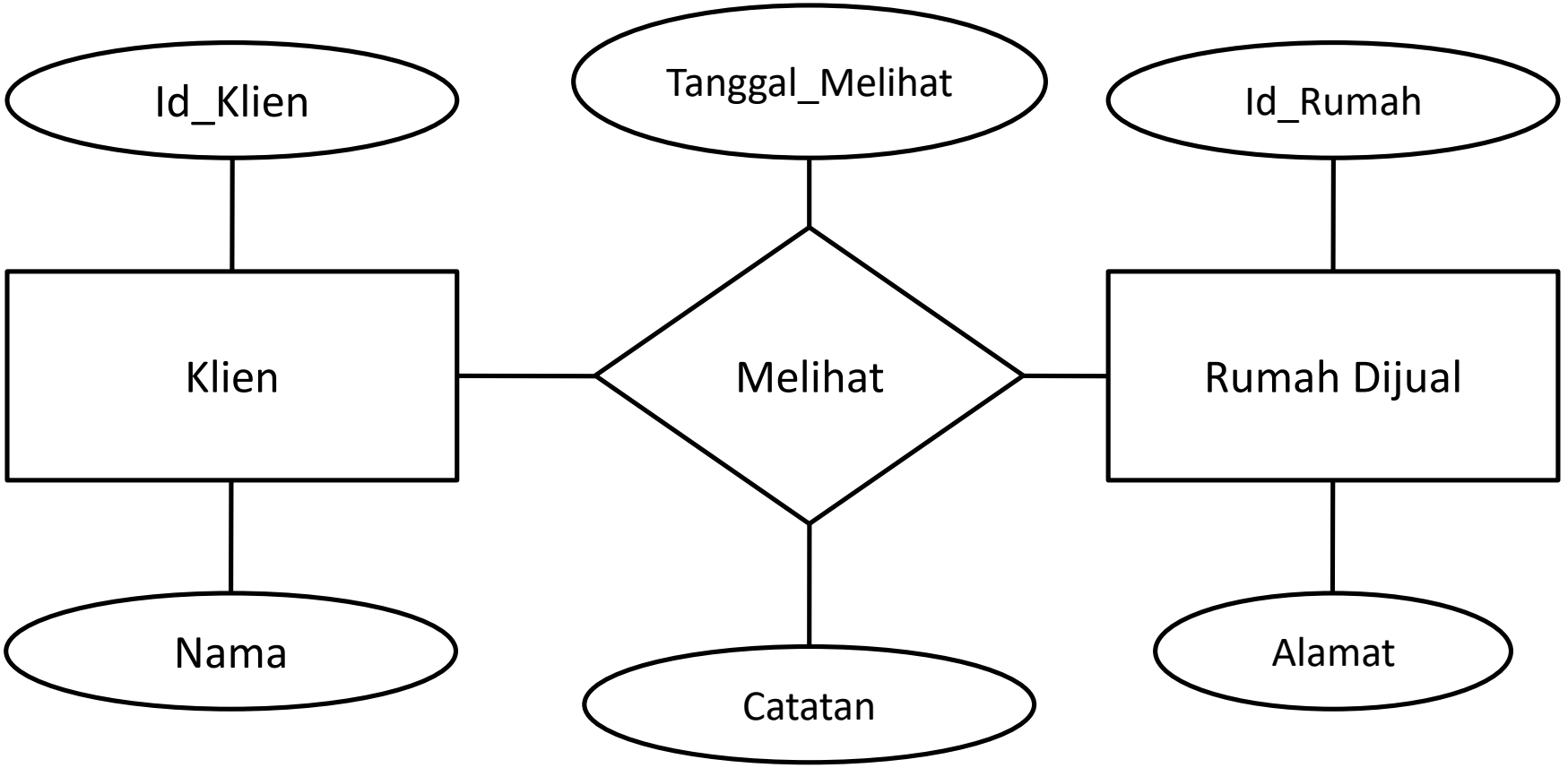


# Relationship with Attributes

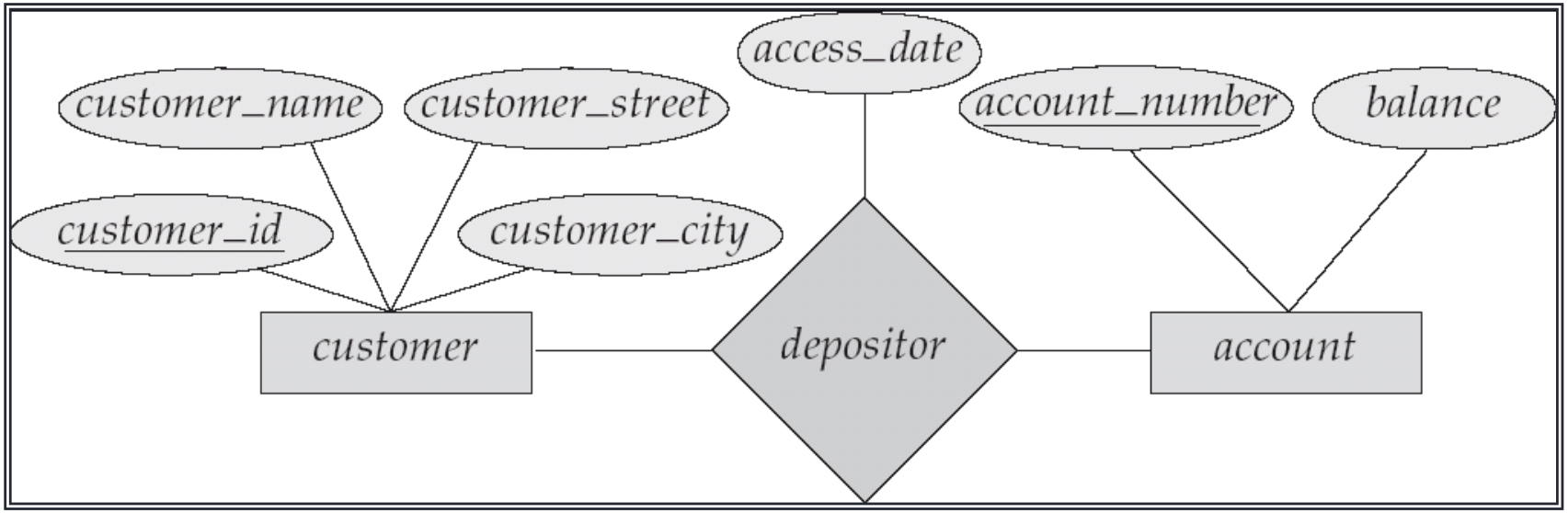
- An attribute can also be property of a relationship set.
- For example, the depositor relationship set between entity sets customer and account may have the attribute access-date



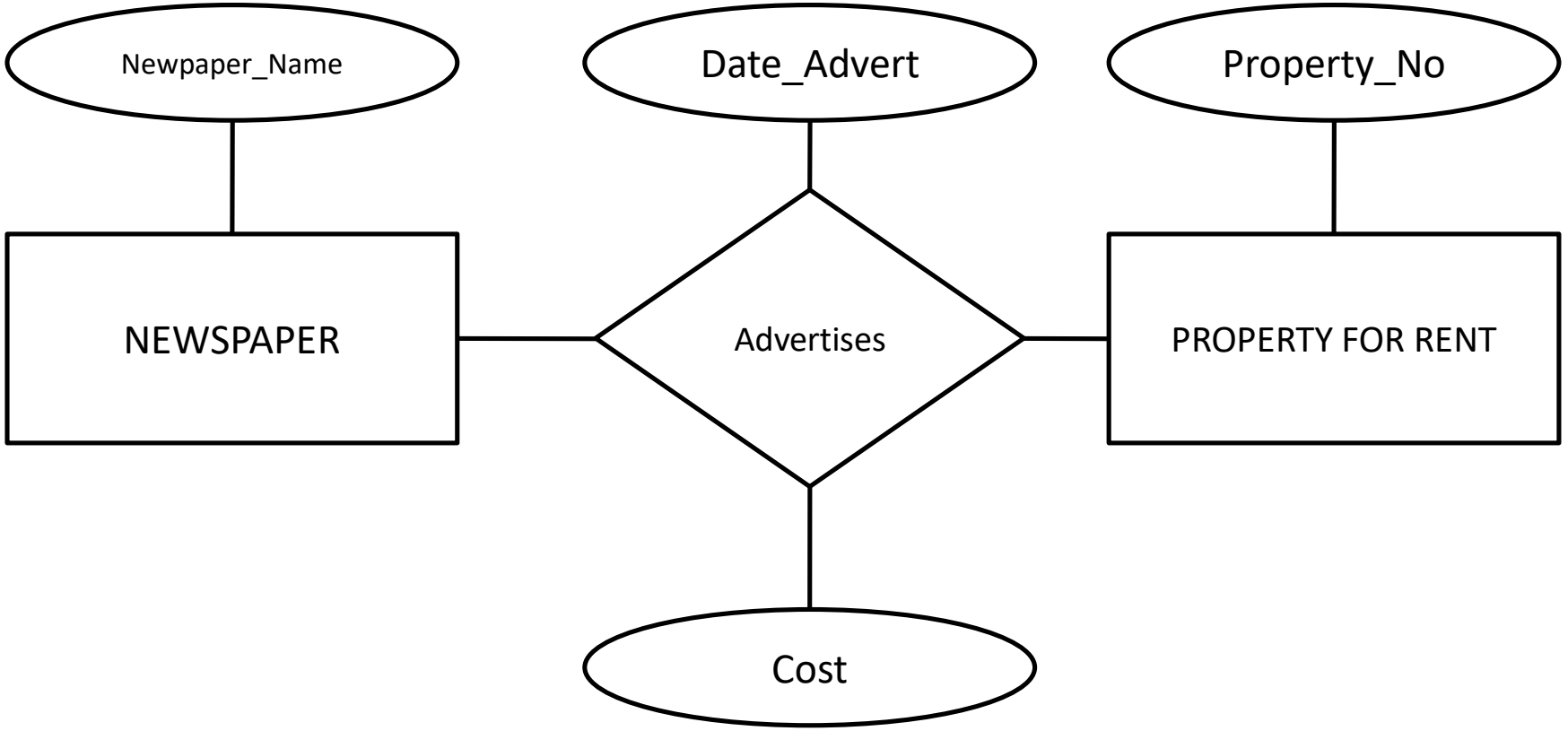
# Relationship with Attributes



# Relationship with Attributes



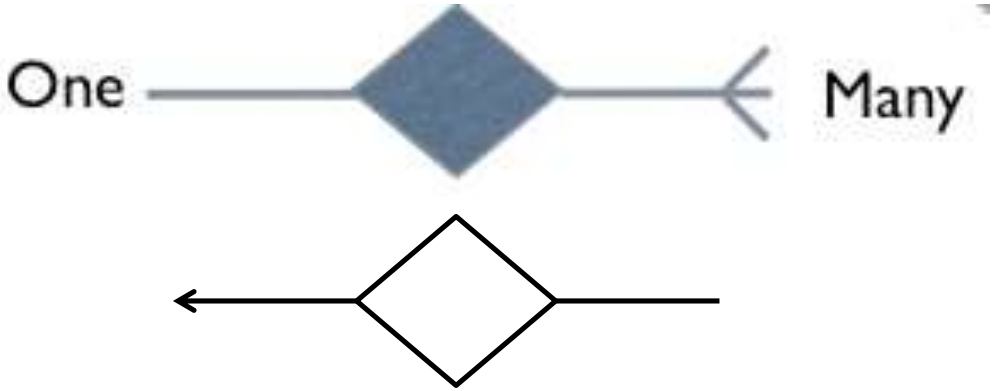
# Relationship with Attributes




An example of a relationship called `Advertises` with attributes `dateAdvert` and `cost`.

# Cardinality Ratios

- Each entity in a relationship can participate in zero, one, or more than one instances of that relationship
- The ends of the link in relationship show cardinality



# Cardinality Ratios

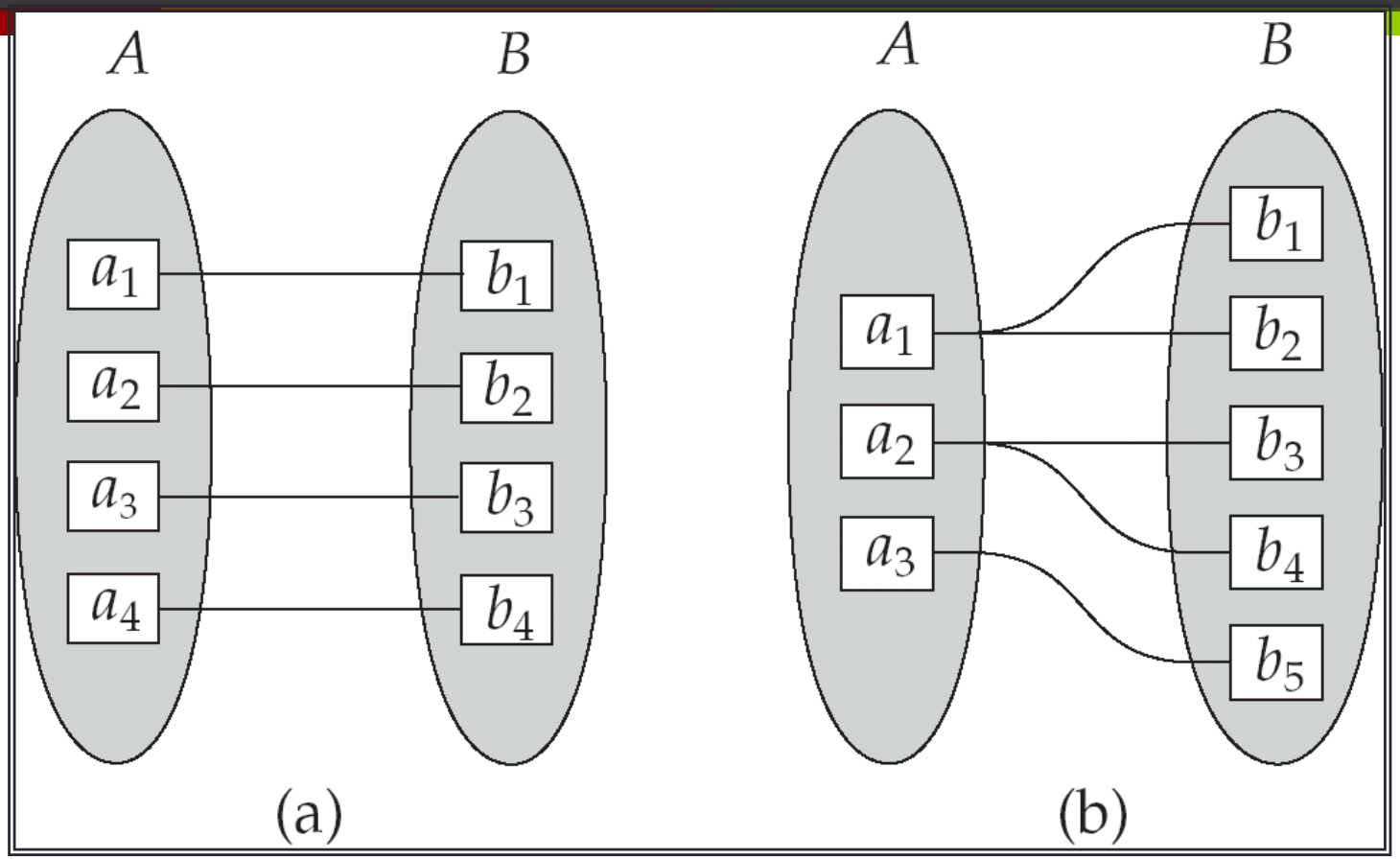
- For a binary relationship set the mapping cardinality must be one of the following types:
  - **One to one**
  - **One to many**
  - **Many to one**
  - **Many to many**

# Cardinality Ratios

- **One to one.** An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A

- **One to many.** An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A.

# Cardinality Ratios



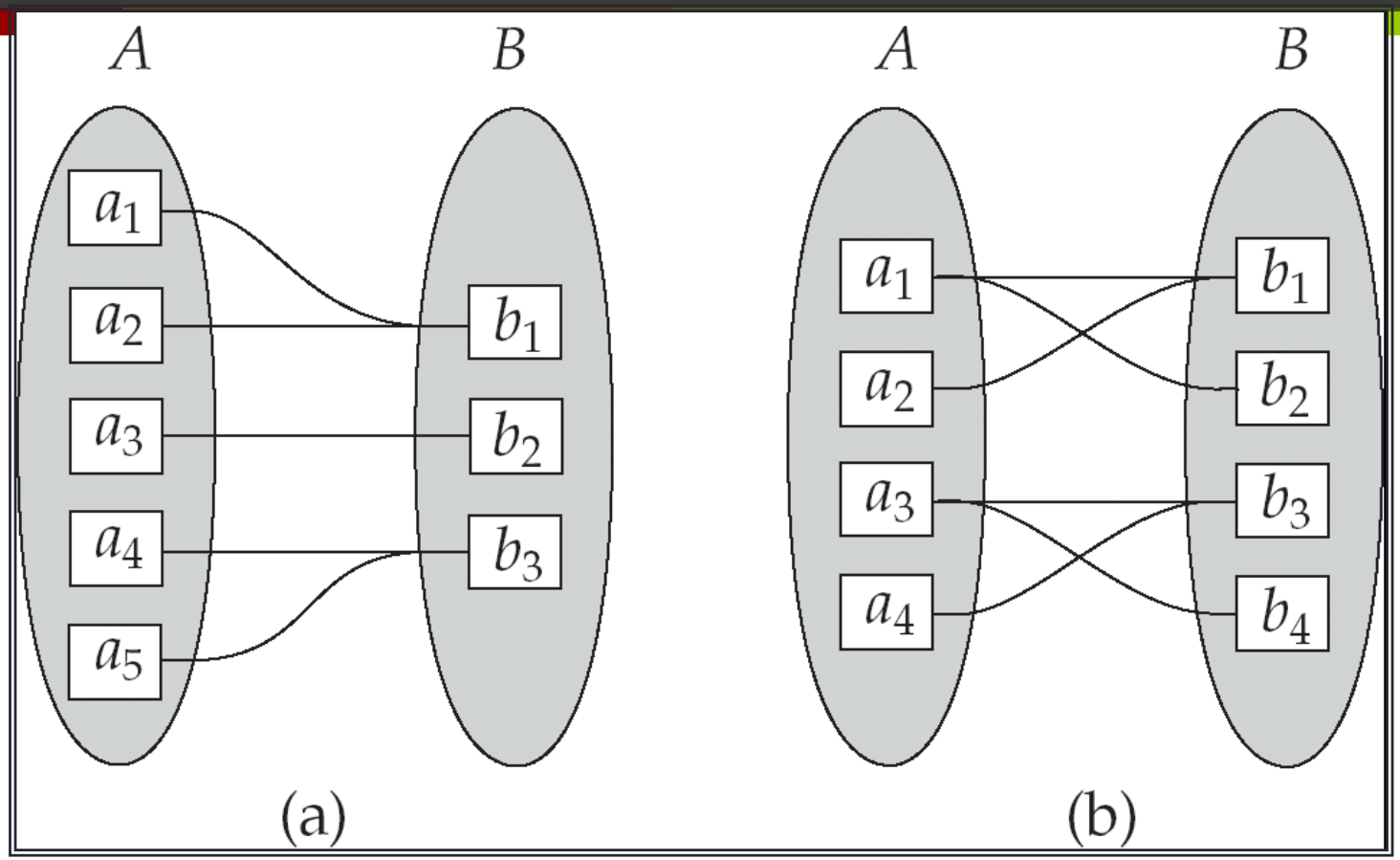

One to one

One to many

# Cardinality Ratios

- **Many to one.** An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A.
- **Many to many**. An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A.

# Cardinality Ratios




Many to one

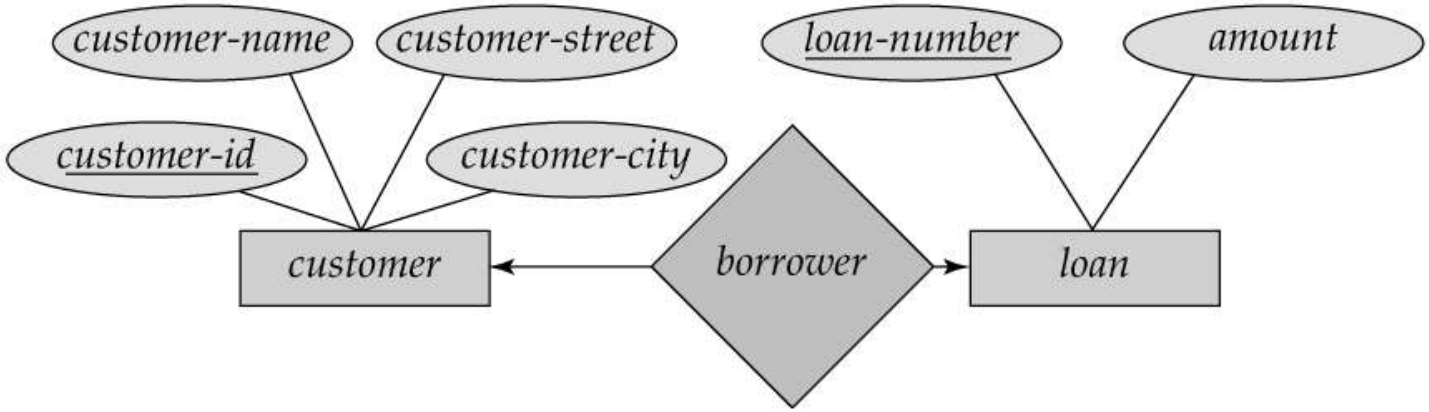
Many to many

# Cardinality Ratios

- To distinguish among relationship types by drawing either
  - a directed line (→), signifying “one,” or
  - an undirected line (—), signifying “many,” between the relationship and the entity

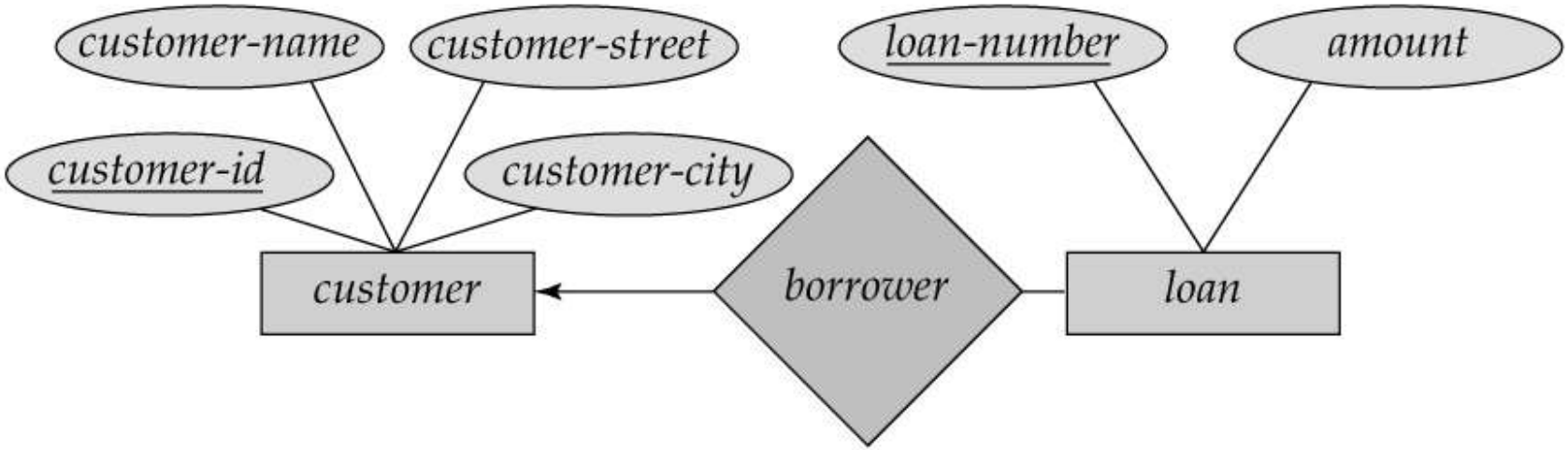
# One-to-one Relationship

- A customer is associated with at most one loan via the relationship *borrower*
- A loan is associated with at most one customer via *borrower*



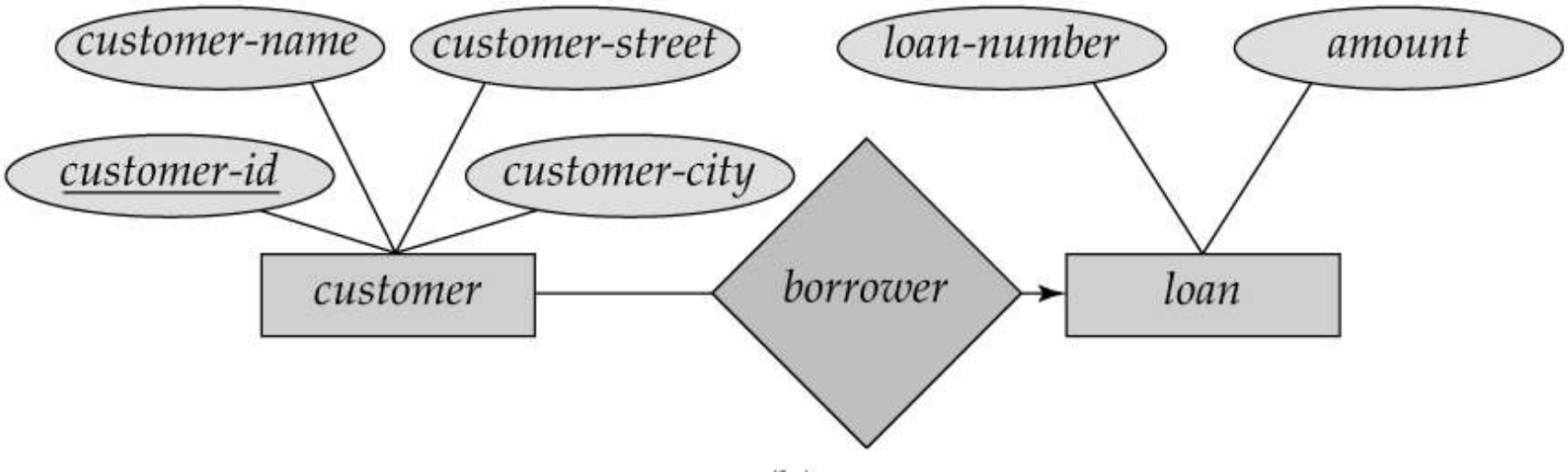
# One-To-Many Relationship

- In the one-to-many relationship a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*



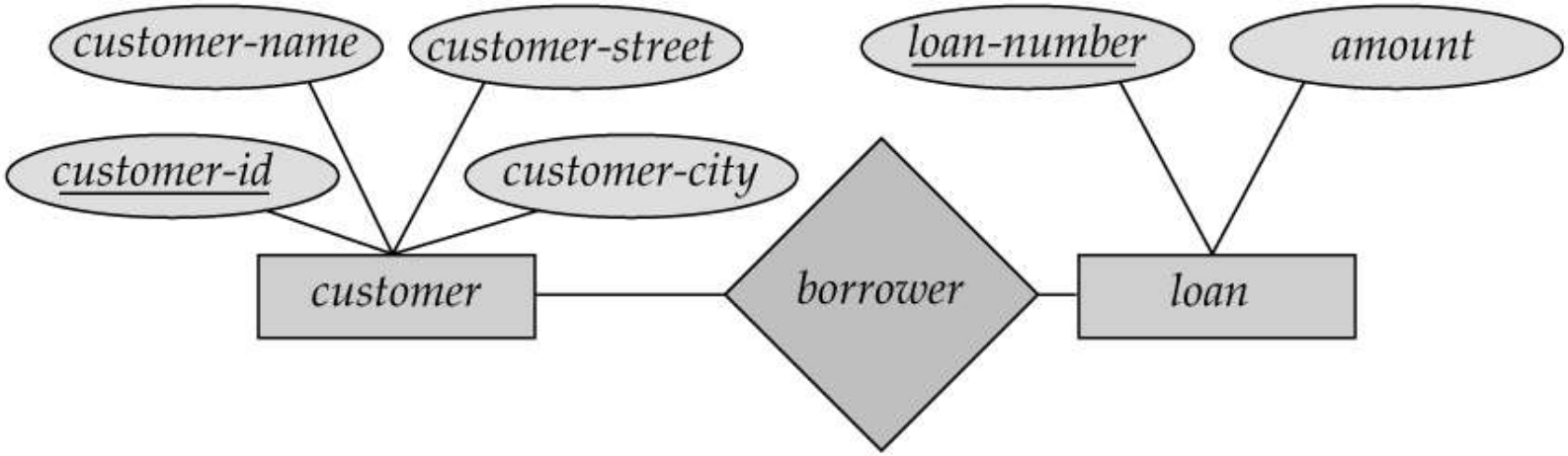
# Many-To-One Relationships

- In a many-to-one relationship a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



# Many-To-Many Relationship

- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower



# Cardinality Constraints

- Jenis kekangan kardinalitas yang diterapkan dalam hubungan, yaitu:
  - Kardinalitas minimum
  - Kardinalitas  maksimum

# Cardinality Constraints

- Kardinalitas Minimum
  - Jumlah tersedikit suatu instans dari suatu tipe entitas yang dapat dikaikan dengan setiap instans pada tipe entitas lain
  - Jumlah minimumnya dapat berjumlah 0 atau 1

# Cardinality Constraints

- Kardinalitas Maksimum
  - Jumlah terbanyak dari instans suatu tipe entitas yang dapat dikaitkan dengan setiap instans dari tipe entitas lain
  - Jumlah maksimumnya berupa 1 atau banyak

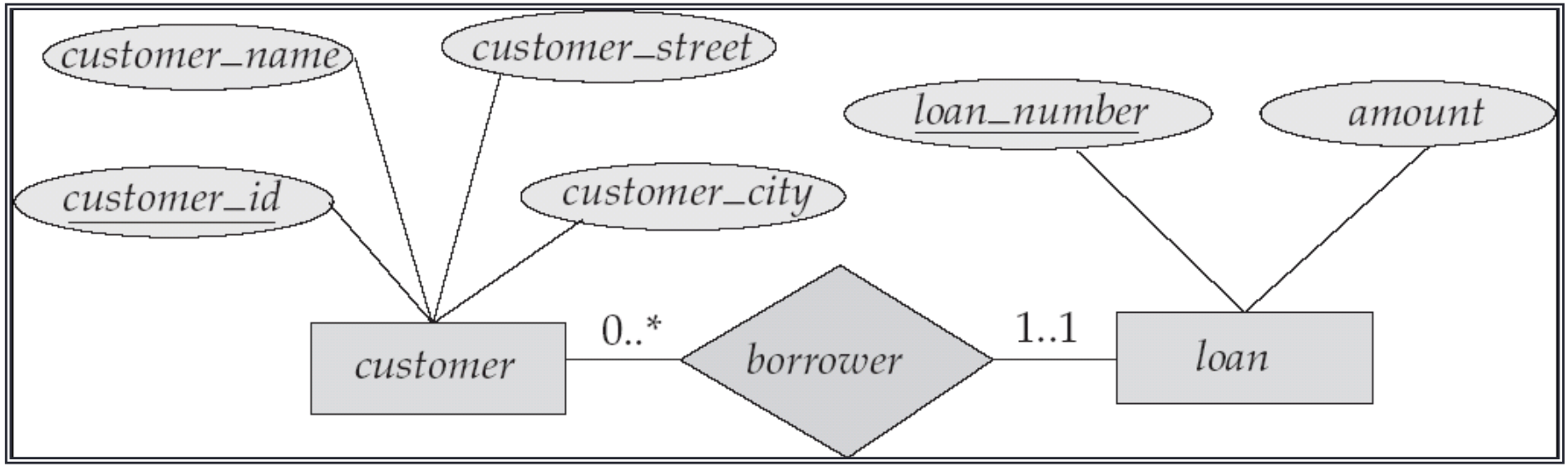
# Cardinality Constraints

- associated minimum and maximum cardinality, shown in the form *l..h*, where
  - *l* is the minimum cardinality
  - *h is* the maximum cardinality

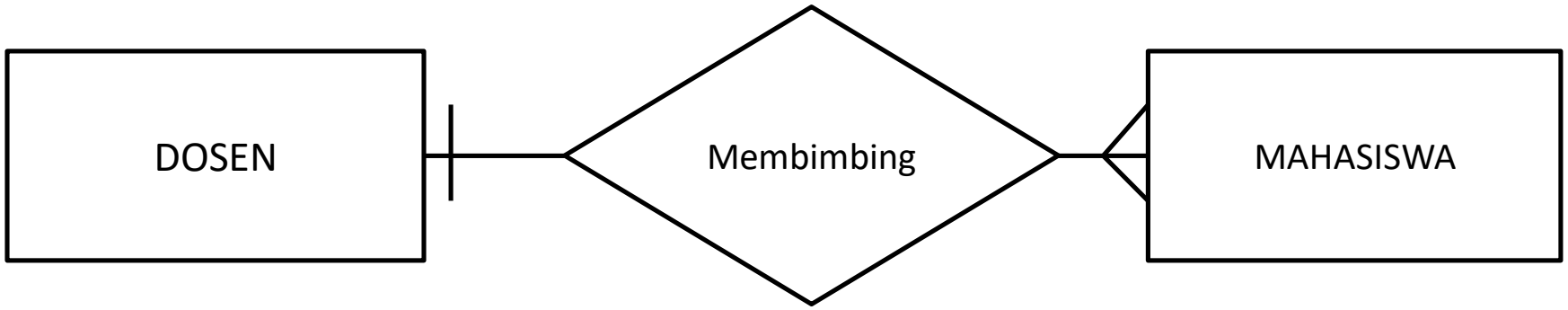
# Cardinality Constraints

- A minimum value of 1 indicates total participation of the entity set in the relationship set.
- A maximum value of 1 indicates that the entity participates in at most one relationship
- while a maximum value $*$ indicates no limit.
- Note that a label $1..*$ on an edge is equivalent to a double line

# Alternative Notation for Cardinality Limits

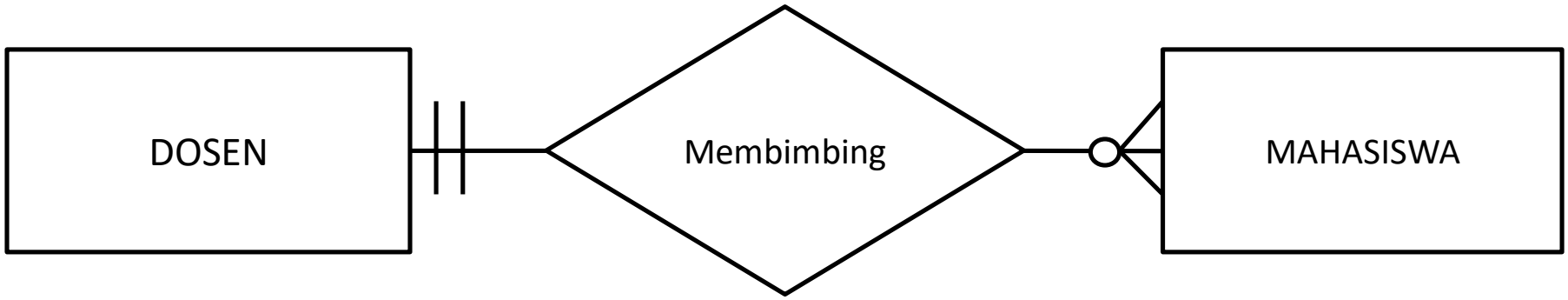


# Cardinality Constraints




Kardinalitas Maksimum

# Cardinality Constraints




Kardinalitas Minimum

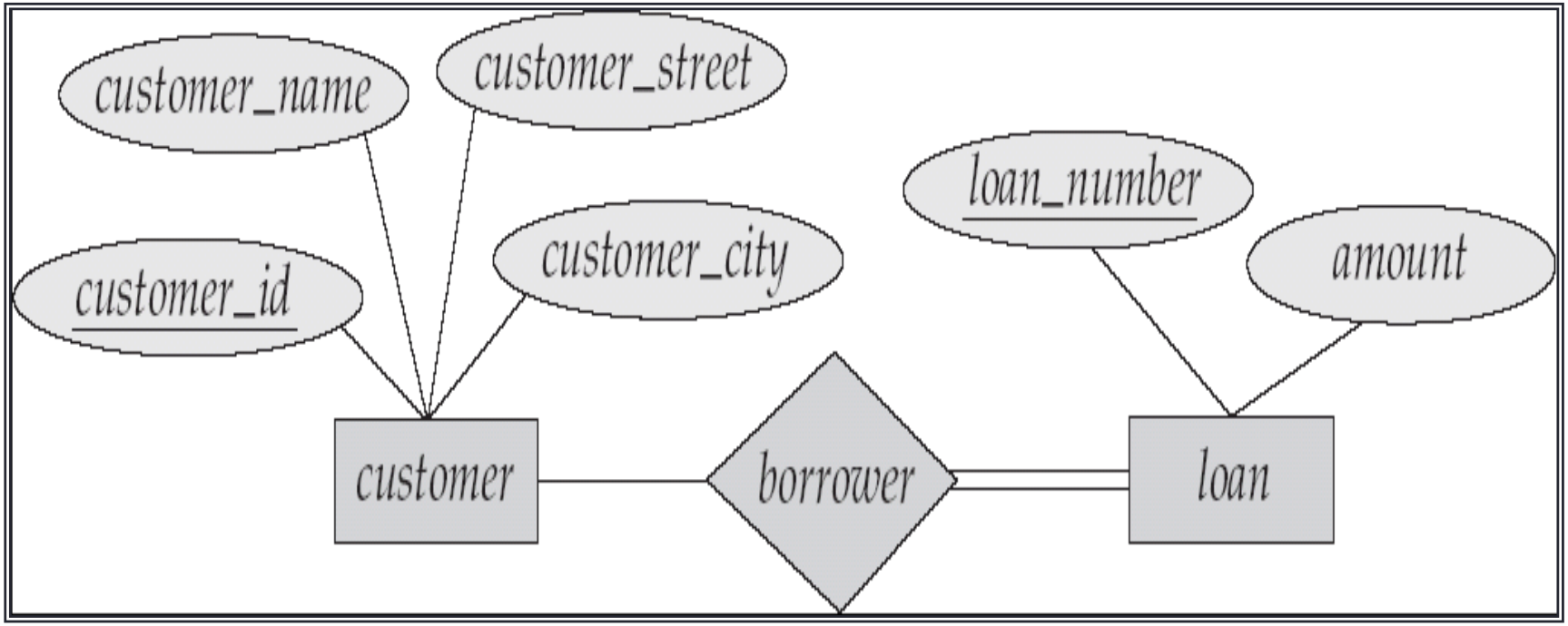
# Participation of an Entity in a Relationship

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship
  - E.g. participation of loan in borrower is total
    - every loan must have a customer associated to it via borrower

# Participation of an Entity in a Relationship

- Partial participation:  some entities may not participate in any relationship in the relationship
  - Example: participation of customer in borrower is partial

# Participation of an Entity in a Relationship



# Strong and Weak Entity

- **Strong entity**
  - An entity that is not existence-dependent on some other entity.
  - each entity occurrence is uniquely identifiable using the primary key attribute(s) of that entity .
    - Example, we can uniquely identify each member of staff using the `staffNo` attribute, which is the primary key for the `Staff` entity type

# Strong and Weak Entity

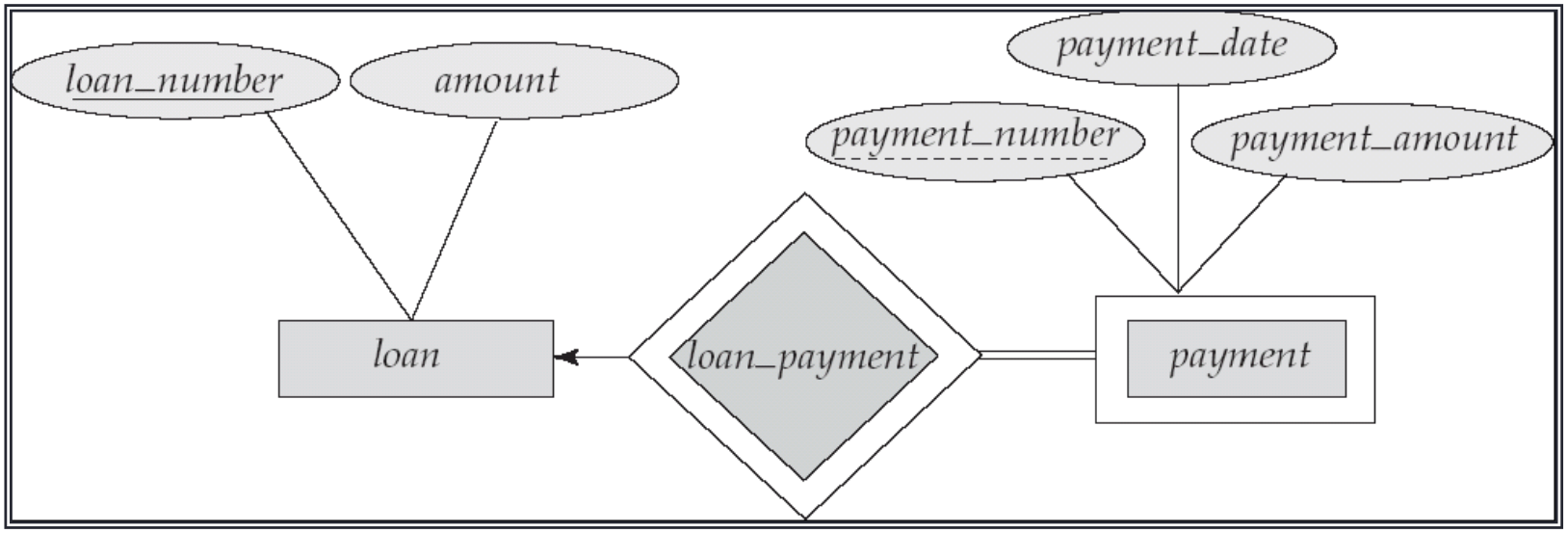
- **Weak entity**
  - An entity that does not have a primary key is referred to as a weak entity.
  - The existence of a weak entity depends on the existence of a identifying entity
    - it must relate to the identifying entity via a total, one-to-many relationship from the identifying to the weak entity
    - Identifying relationship depicted using a double diamond

# Strong and Weak Entity

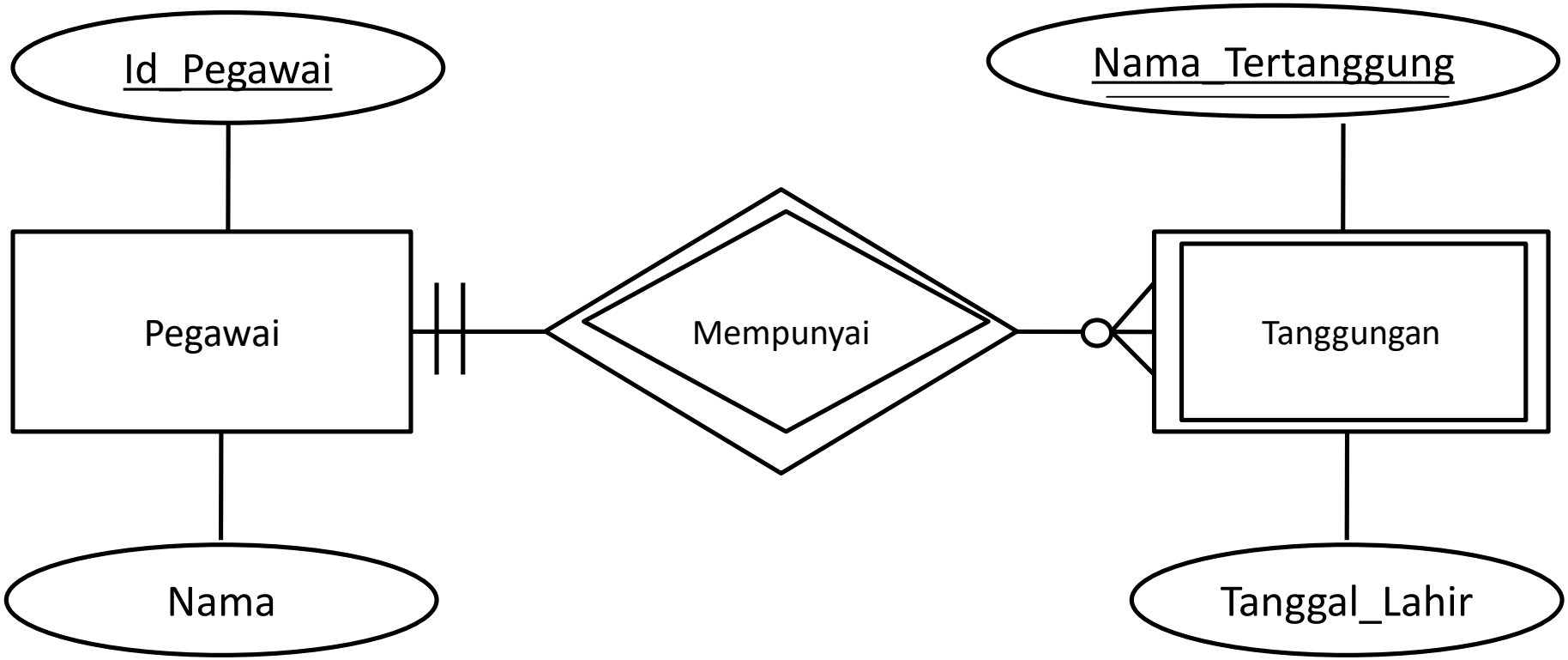
- **Weak entity**
  - The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
  - The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

# Weak Entity

- We depict a weak entity by double rectangles.
- We underline the discriminator of a weak entity set with a dashed line.
- payment_number – discriminator of the payment entity set
- Primary key for payment – (loan_number, payment_number)



# Strong and Weak Entity Example




Hubungan entity setkuat dan entity set lemah

# Strong and Weak Entity Example

- Entity set kuat
  - Pegawai
- Entity set lemah
  - Tanggungan
- Hubungan pengidentifikasi
  - Mempunyai
- Pengenal parsial (kunci parsial)
  - Nama_Tertanggung
- Kunci primer Tanggungan
  - Id_Pegawai, Nama_Tertanggung

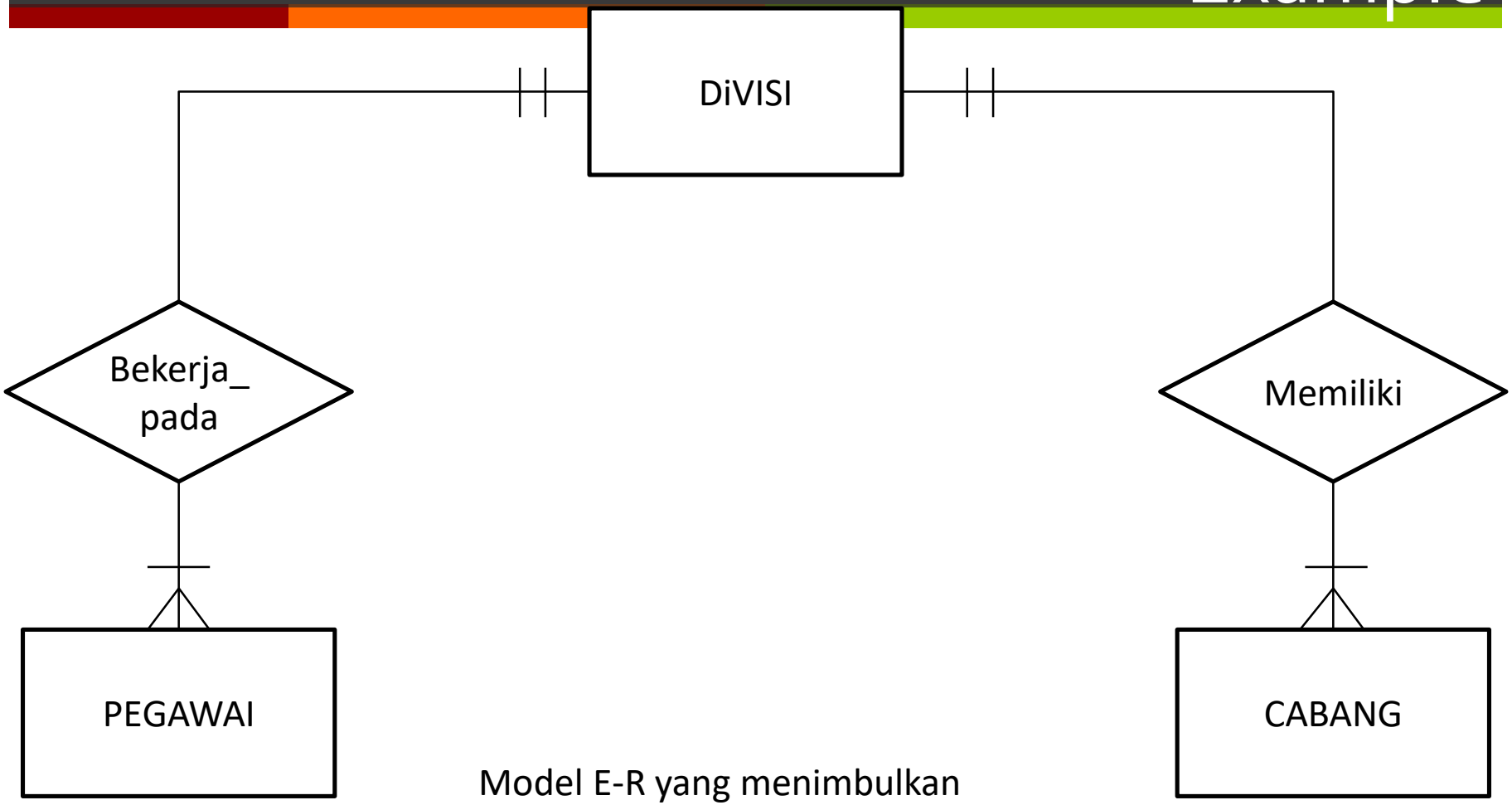
# Problems in E-R Model

- **Jebakan koneksi (Connection traps)**
  - Terjadi karena kesalahan dalam menggambarkan hubungan.
- Dua jenis connection traps
  - **Fan traps**
  - **Chasm traps**

# Fan traps

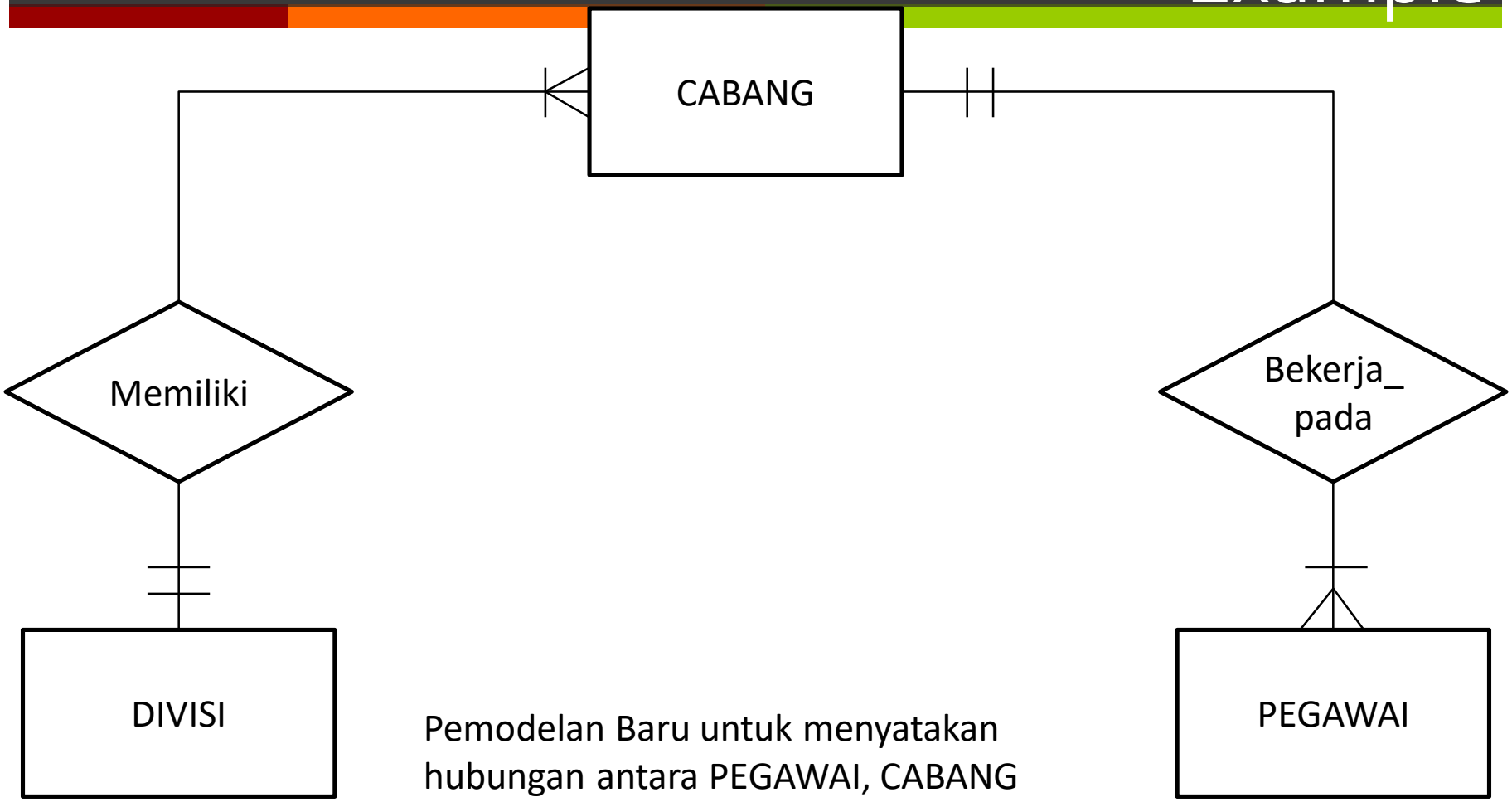
- Suatu jebakan yang membuat hubungan antara instance entities menjadi rancu
- Dua atau lebih hubungan bersifat 1:M bermuara pada sebuah entity set dan terjadi kekeliruan dalam menggambarkan hubungan.

# Fan traps Example




Model E-R yang menimbulkan
Fan Trap

# Fan traps Example

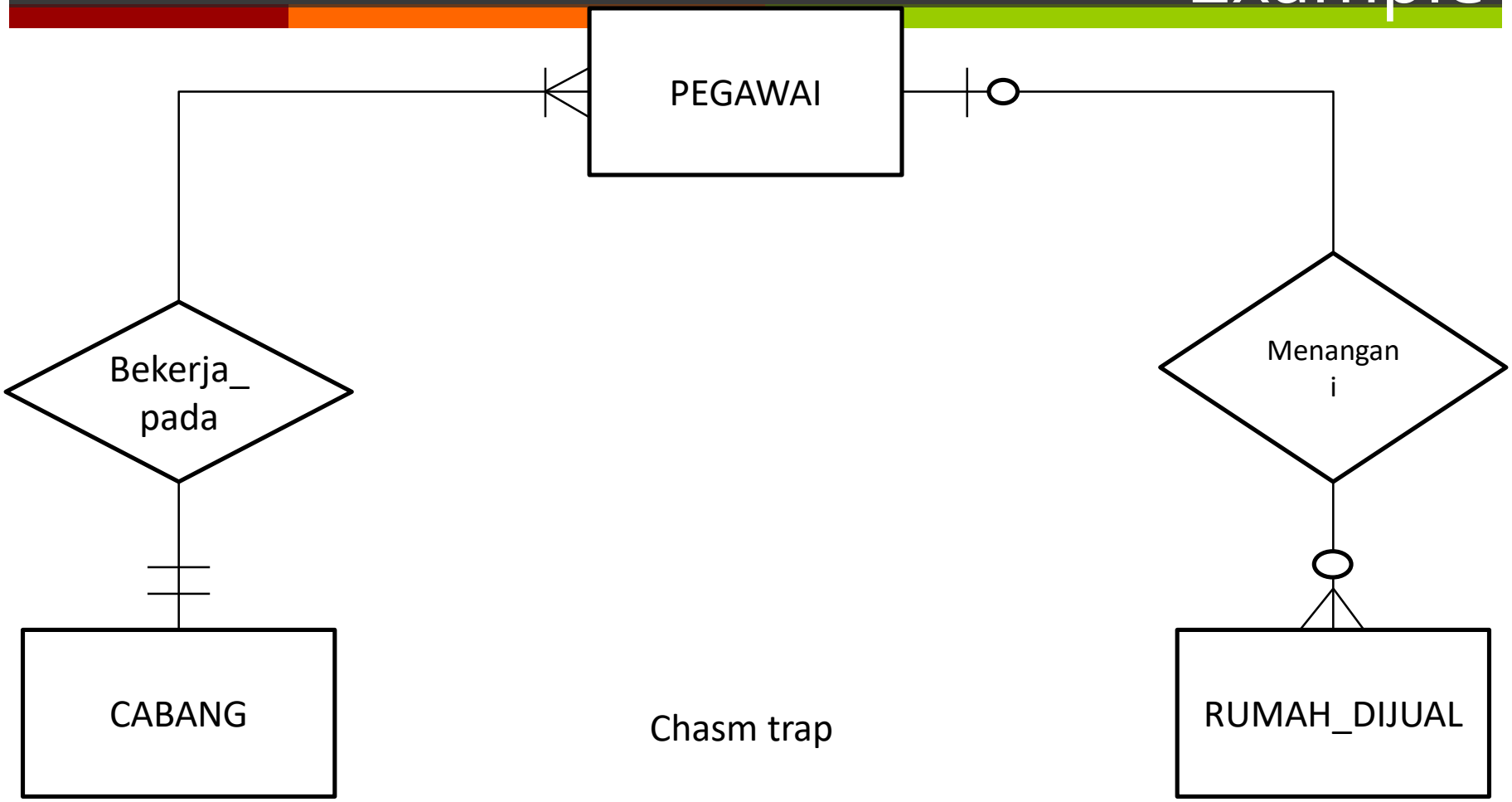



Pemodelan Baru untuk menyatakan hubungan antara PEGAWAI, CABANG dan DIVISI

# Chasm traps

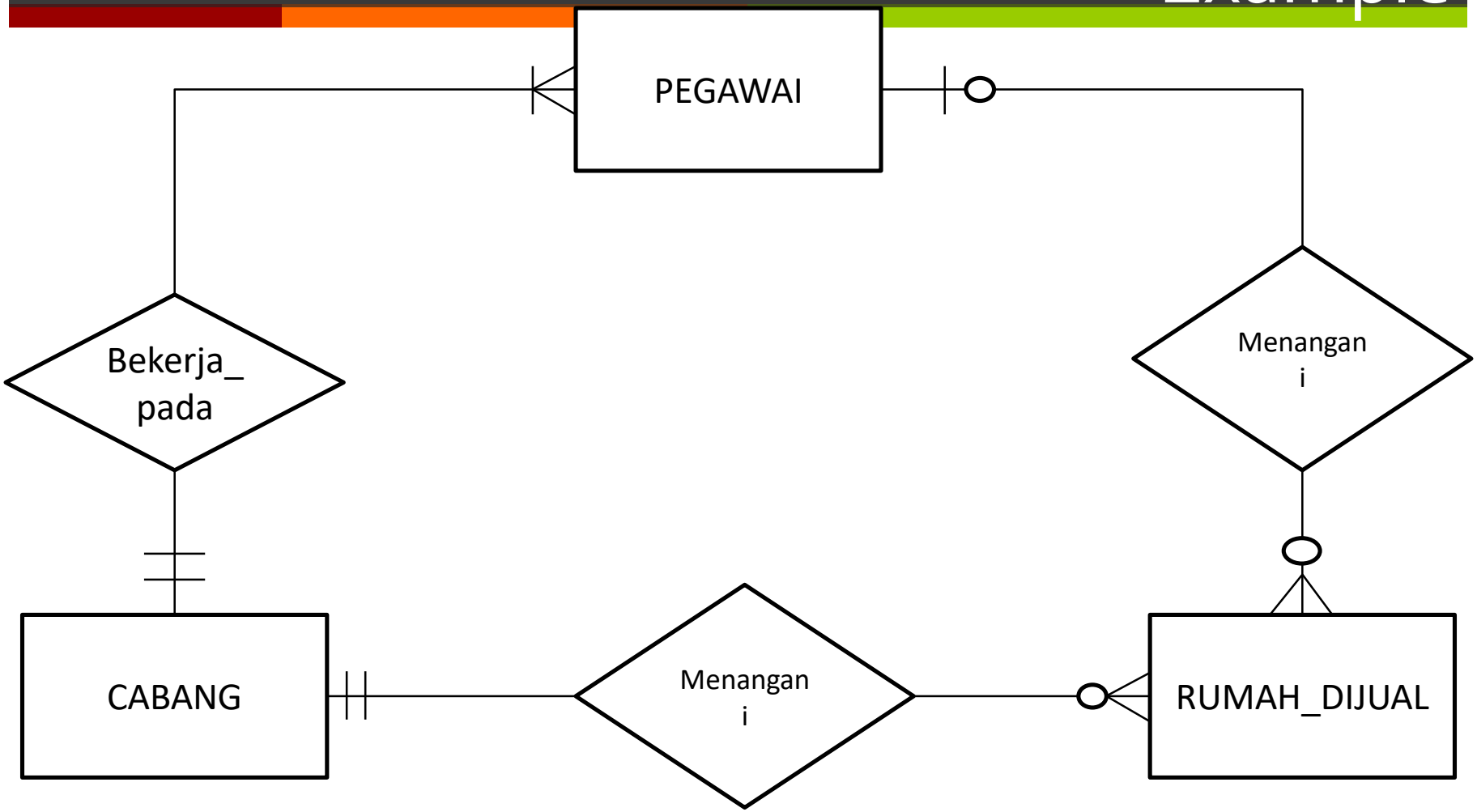
- Suatu jebakan yang membuat instance entity tertentu kehilangan hubungan.
- Terjadi karena adanya kekurangan hubungan

# Chasm traps Example




Chasm trap

# Chasm traps Example




Penambahan hubungan untuk mengatasi Chasm trap