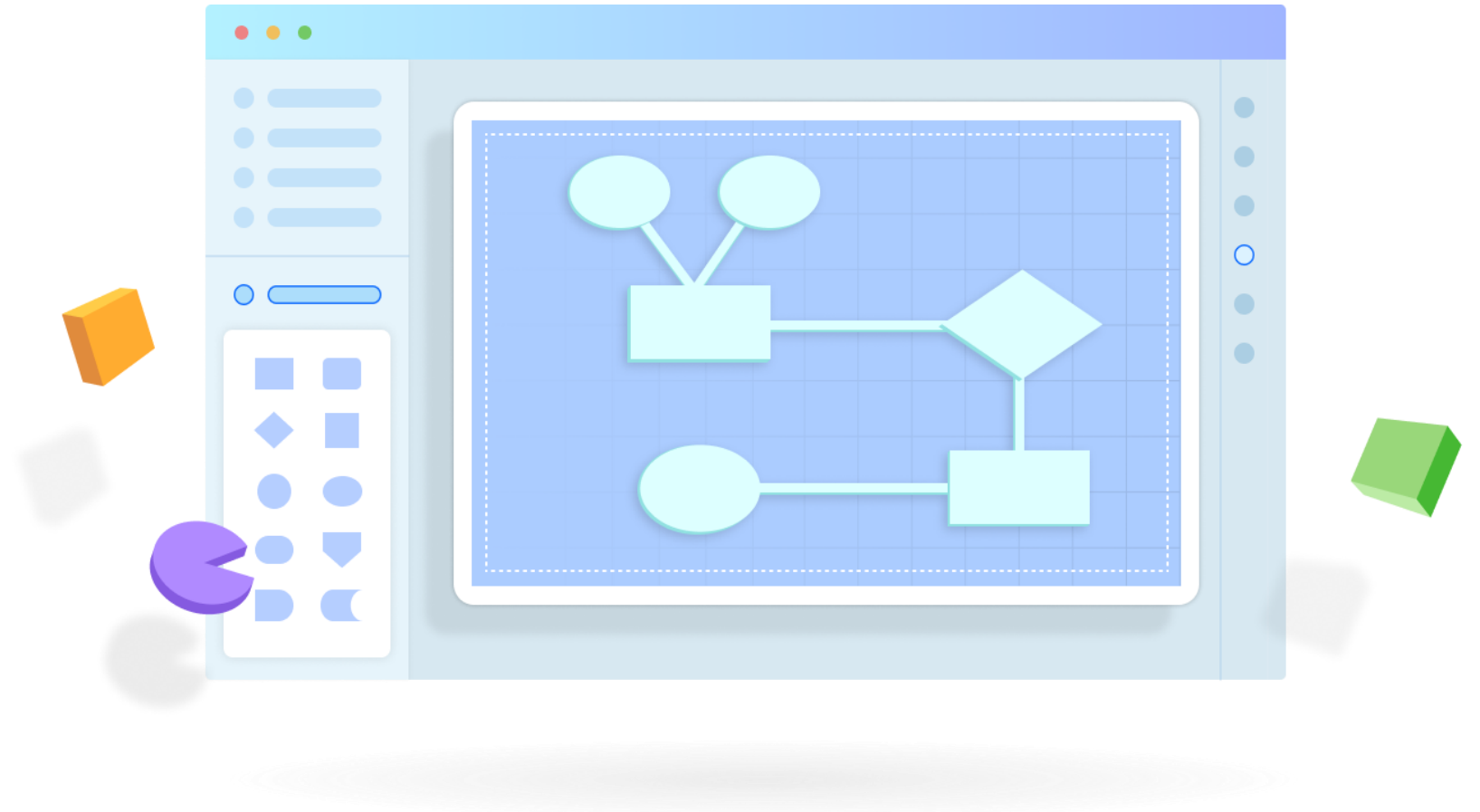




# PEMODELAN ERD



# OUTLINE

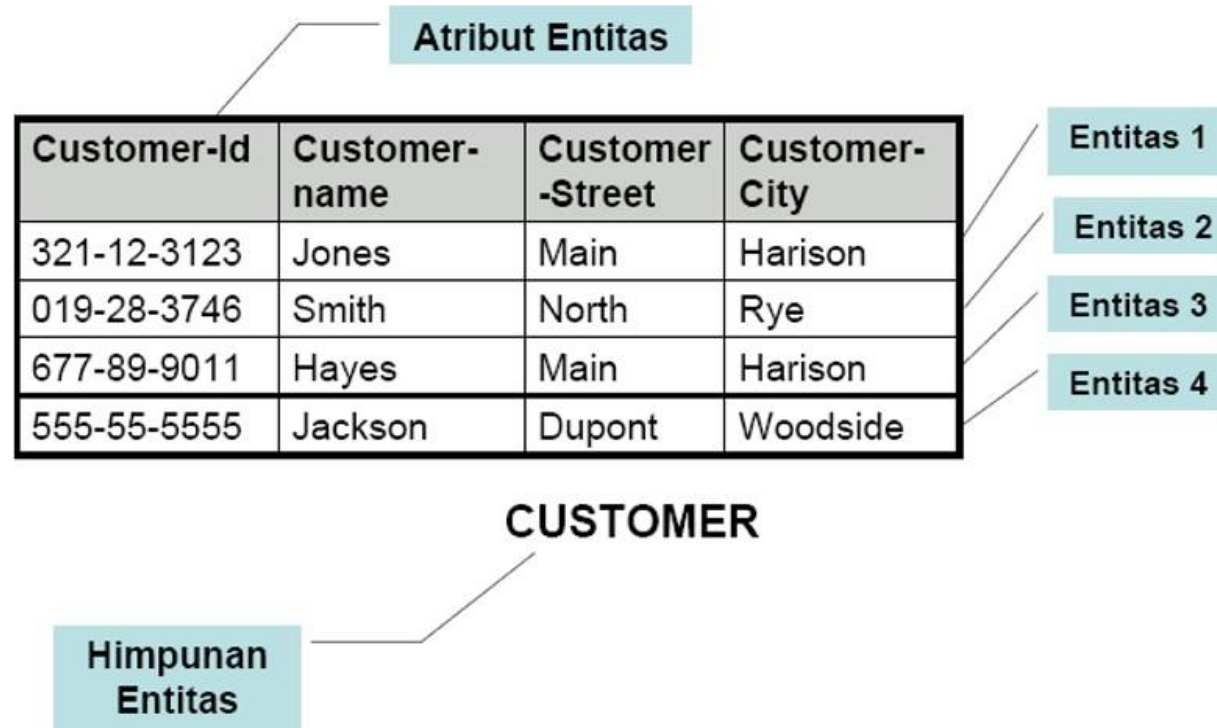
- **Komponen (Entitas, Atribut, Relationship)**
- **Pemetaan Derajat Relasi**
- **Kunci**
- **ER Diagram**

# Komponen - Entitas

- Entitas adalah “sesuatu” atau “objek” di dunia nyata yang dapat dibedakan dari objek lain  
Contoh : Orang tertentu (Mahasiswa), Kursi, Buku
- Entitas bisa berupa :
  - Entitas kongkrit, Contoh : Mahasiswa, Buku
  - Entitas abstrak, Contoh : Pinjaman, Jadwal, Nilai
- Entitas memiliki atribut  
Contoh : Mahasiswa memiliki Nim, nama & Alamat
- Himpunan Entitas (Entity Set): Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama
- Entitas menunjuk kepada pada individu suatu objek sedangkan himpunan entitas menunjuk pada rumpun (family) dari individu tersebut
- Contoh : Semua orang yang memiliki rekening di Bank (nasabah), Semua Pelanggan, Mahasiswa, Dokter

# Komponen - Entitas Lanj.

Contoh Himpunan Entitas Customer:



# Komponen - Atribut

- Setiap Entitas memiliki atribut yang mendeskripsikan karakteristik (properti) dari entitas tersebut
- Contoh : Customer = (Customer-Id, Customer-Name, Customer-Street, Customer-City)
- Setiap atribut akan memiliki nilai (values)
- Domain (Value Set) merupakan batas-batas nilai yang diperbolehkan bagi suatu atribut
- Tipe-tipe Atribut
  1. Simple dan Composite attributes
    - Atribut Simple : Atribut sederhana yang tidak dapat dibagi dalam beberapa bagian;
    - Atribut Komposit : Atribut yang dapat dibagi lagi dalam beberapa bagian; contoh : Nama; yang terdiri dari Nama depan dan Nama Belakang

## Komponen - Atribut Lanj.

### 2. Single-valued dan multi-valued attributes

- Atribut Single-valued : Atribut yang memiliki paling banyak satu nilai untuk setiap baris data
- Multi-valued attributes : Atribut yang dapat diisi dengan lebih satu nilai tetapi jenisnya sama. Contoh : NomorTelp, Alamat

### 3. Derived attributes

- Atribut Turunan : Atribut yang diperoleh dari pengolahan dari atribut lain yang berhubungan. Contoh : Umur, IP

### 4. Atribut Mandatory dan Non Mandatory

- Atribut Mandatory adalah atribut yang harus diisi tidak boleh kosong (not null)
- Atribut Non mandatory adalah atribut yang boleh kosong(null).

# Komponen - Relationship

- Relasi adalah hubungan antara beberapa entitas

Contoh : Hayes                      borrower                      L-23

Entitas customer   Relasi                      Entitas account

- Himpunan relasi adalah Kumpulan semua relasi yang merupakan relasi matematik antara  $n \geq 2$  entitas, dari himpunan-himpunan entitas yang ada
- $\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$   
dimana :  $(E_1, E_2, \dots, E_n)$  adalah Entitas,  $(e_1, e_2, \dots, e_n)$   
adalah relasi
- Contoh :  $(\text{Hayes}, \text{L-23}) \in \text{borrower}$



Customer-Id	Customer-name	Customer-Street	Customer-City	Loan-Number	Amount
321-12-3123	Jones	Main	Harison	L-17	1000
019-28-3746	Smith	North	Rye	L-23	2000
677-89-9011	Hayes	Main	Harison	L-15	1500
555-55-5555	Jackson	Dupont	Woodside	L-14	500
				L-19	900

# Relationship

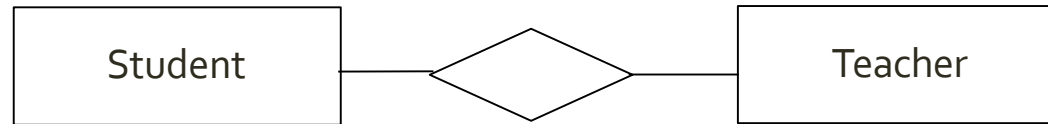
## Komponen – Relationship Lanj.

- Derajat Relasi menunjukkan banyaknya himpunan entitas yang salingberelasi.
- Himpunan relasi melibatkan dua himpunan entitas disebut Binary (atau ber-derajat 2). Secara umum himpunan relasi dalam sistem basis data adalah binary
- Himpunan relasi memungkinkan untuk melibatkan dua himpunan entitas.
- Relasi antara lebih dari dua entitas jarang terjadi.
- Contoh : Jika employee suatu Bank boleh memiliki pekerjaan (Job) pada beberapa cabang (Branch) dengan pekerjaan yang berbeda pada cabang yang berbeda. Maka akan terjadi relasi ternary (berderajat-3) antara himpunan entitas employee, Job dan Branch

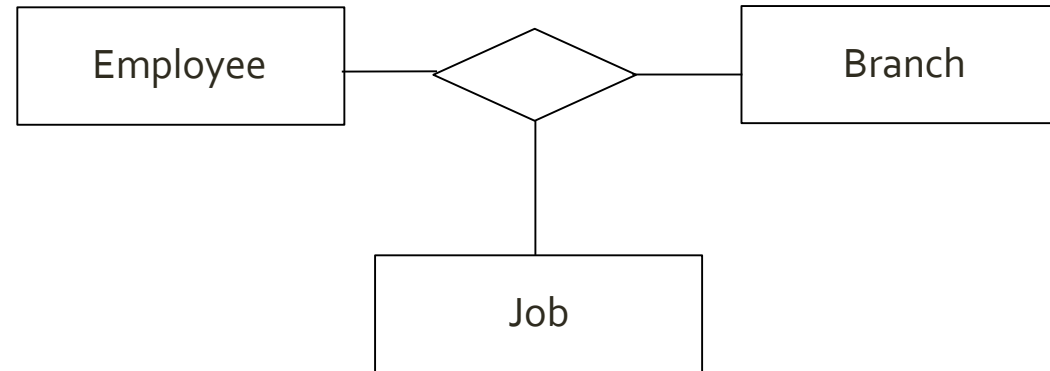
# Komponen – Relationship Lanj.

## Contoh Derajat Relasi

- Binary Relationship



- Ternary Relationship

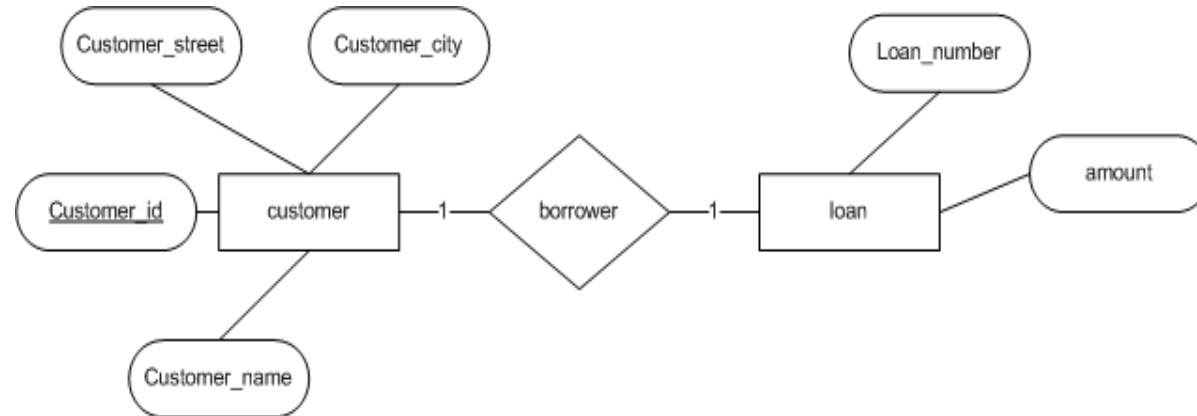


# Pemetaan Derajat Relasi

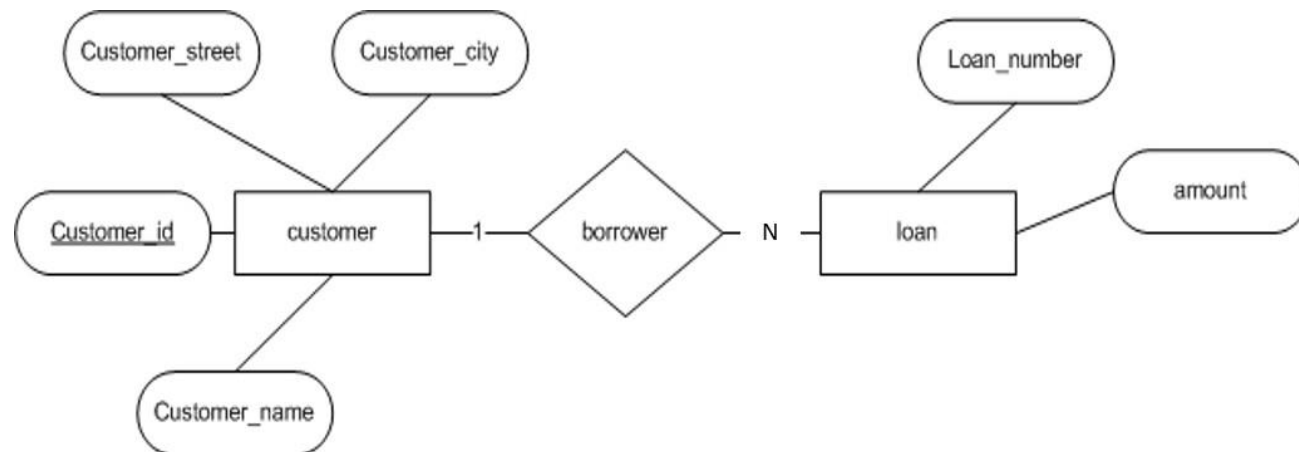
- Menggambarkan banyaknya jumlah maksimum entitas dapat berelasi dengan entitas pada himpunan entitas yang lain.
- Paling banyak digunakan dalam menjelaskan Himpunan relasi biner
- Untuk Himpunan relasi biner pemetaan kardinalitasnya dapat merupakan salah satu dari tipe2 berikut :
  1. Satu ke Satu (One to one)
  2. Satu ke Banyak (One to many)
  3. Banyak ke Satu (Many to one)
  4. Banyak ke Banyak (Many to many)
- Kardinalitas pemetaan dinyatakan dengan 2 cara :
  - a. [Korth] garis berarah (1) dan garis tidak berarah (Banyak).
  - b. [Date] menuliskan kardinalitasnya pada garis

# Pemetaan Derajat Relasi Lanj.

Satu ke Satu (One to one)

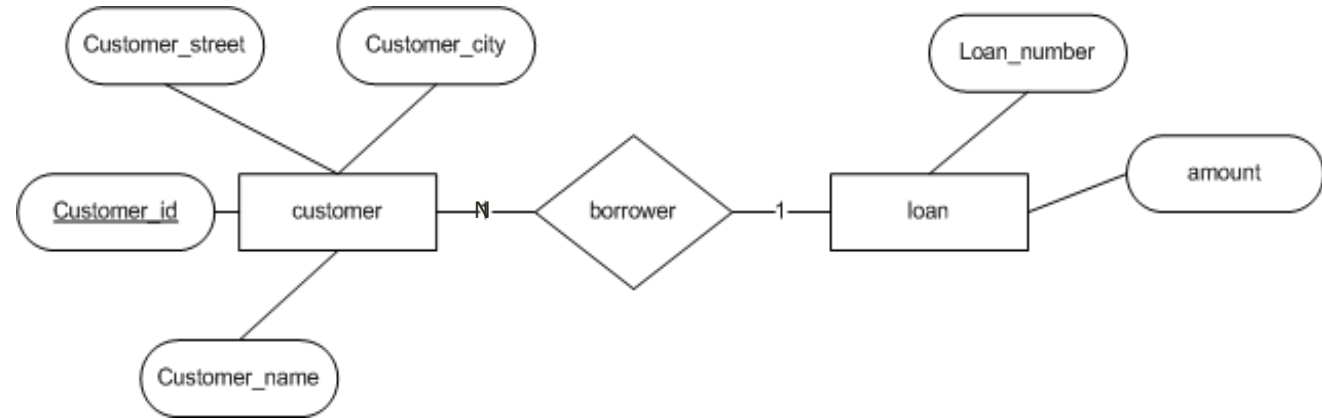


Satu ke Banyak (One to Many)

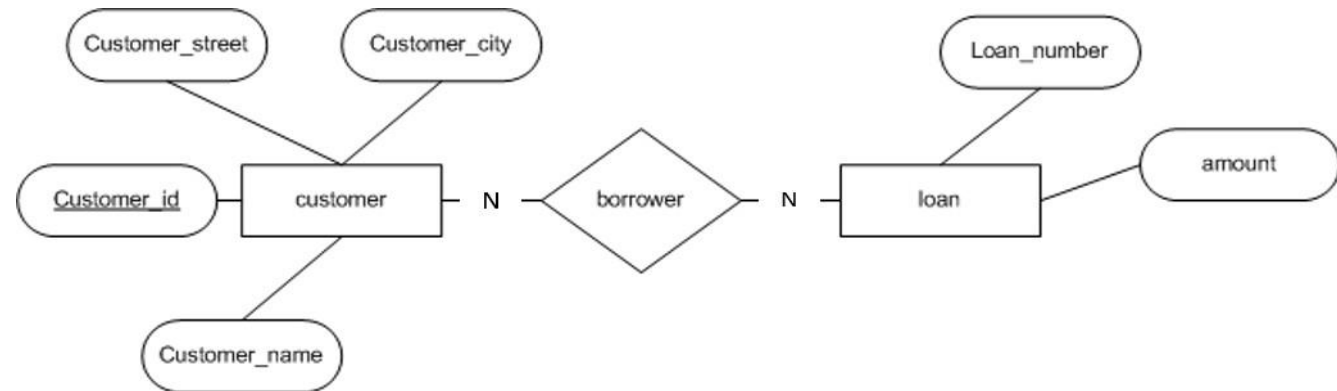


# Pemetaan Derajat Relasi Lanj.

Banyak ke Satu (Many to one)



Banyak ke Banyak (Many to Many)



# Kunci

- Penggunaan key merupakan cara untuk membedakan suatu entitas didalam himpunan entitas dengan entitas lain
- Secara konsep, masing-masing entitas (nilainya) berbeda, perbedaannya terlihat pada isi dari masing-masing atributnya
- Oleh karena itu, dibutuhkan suatu atribut yang memiliki nilai yang menjadi pembeda dengan entitas lain
- Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua row dalam relasi secara unik
- Ada 3 macam key yang dapat diterapkan pada suatu relasi
  1. Super Key merupakan seluruh atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah relasi secara unik.

## Kunci (Lanj.)

2. Candidate Key merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah relasi secara unik
3. Primary Key merupakan salah satu dari candidate key yang terpilih

Pemilihan primary key dari sejumlah candidate key umumnya didasari oleh :

- Key tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan
- Key tersebut lebih ringkas
- Jaminan keunikan key tersebut lebih baik

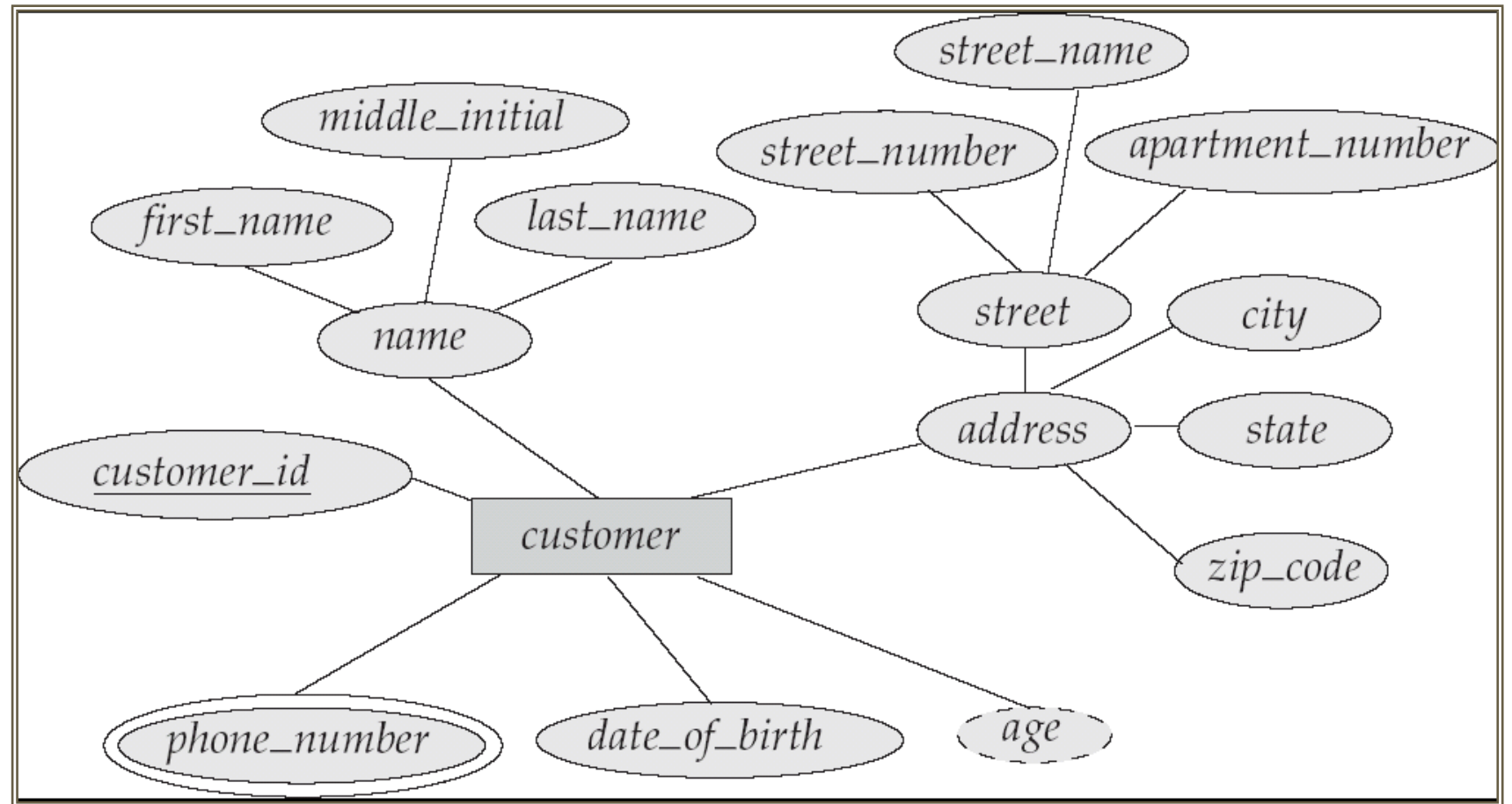


# Diagram ER

- Diagram ER merupakan model konseptual untuk menggambarkan struktur logis dari basis data berbasis grafis
- Diagram ER dibangun dari komponen berikut :
  - Segi Empat : menunjukkan Entitas
  - Elips : menunjukkan Atribut
  - Diamond : menunjukkan Relasi
  - Garis : Link yang menghubungkan antara Entitas dengan atribut, dan entitas dengan relasi
  - Elips dobel : Menunjukkan atribut yang multivalued
  - Elips dengan garis terputus : Menunjukkan atribut turunan

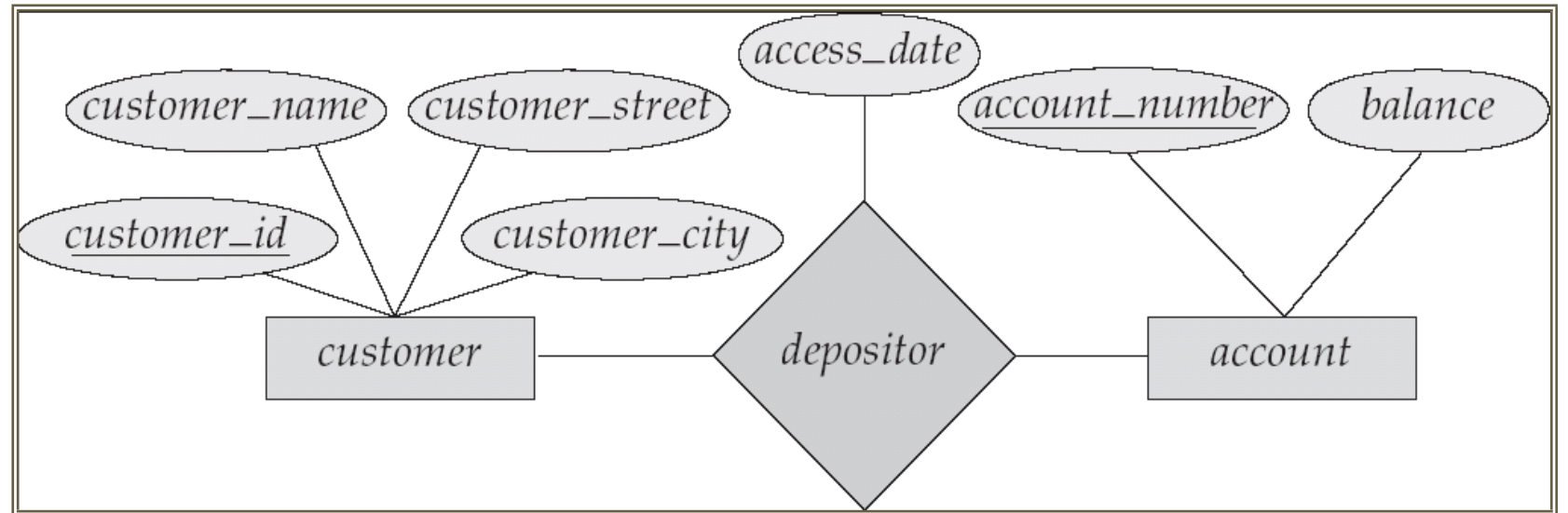
## Diagram ER (Lanj.)

Diagram E-R dengan atribut Composite, Multivalued dan derived



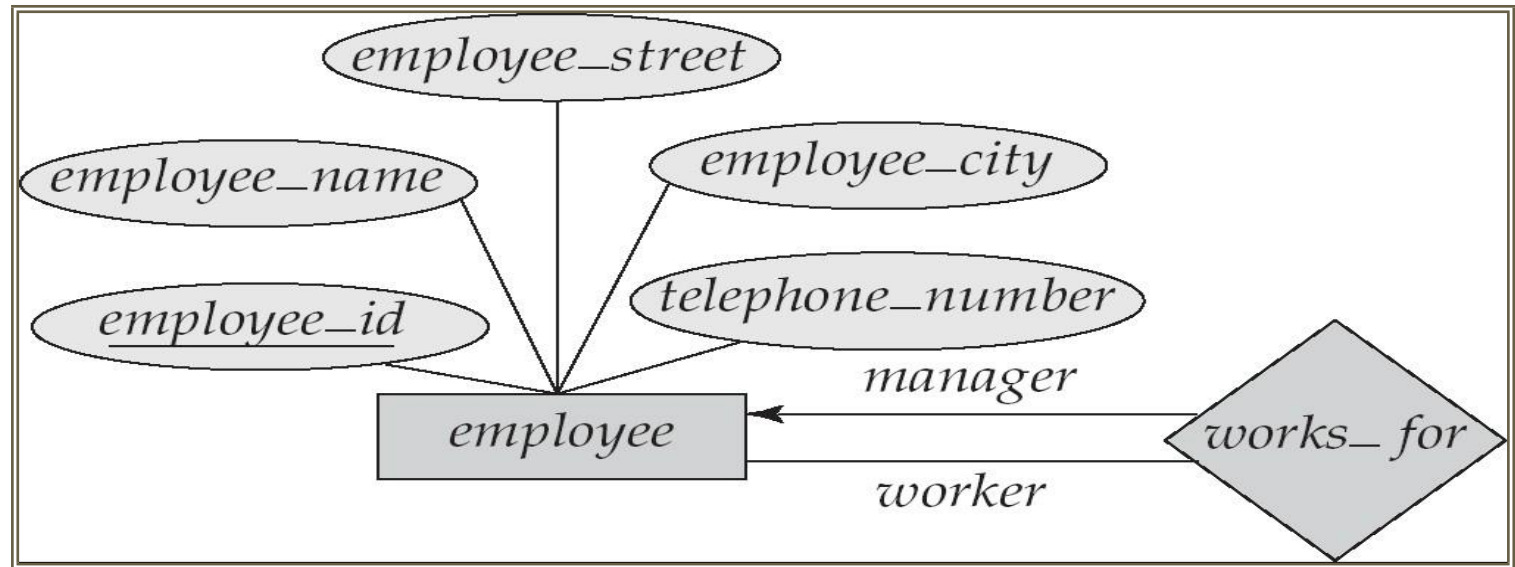
## Diagram ER (Lanj.)

### Himpunan Relasi dengan Atribut



## Peran (Roles)

- Relasi Himpunan entitas tidak harus dalam bentuk yang berbeda
- Label “manager” dan “worker” disebut Roles (peran), yang menspesifikasi bagaimana entitas employee berinteraksi melalui relasi Works-for
- Peran dalam ER diagram diindikasikan dengan memberikan label (nama) pada garis yang menghubungkan relasi dengan entitas
- Label peran bersifat optional dan digunakan untuk mengklarifikasi semantik suatu relasi

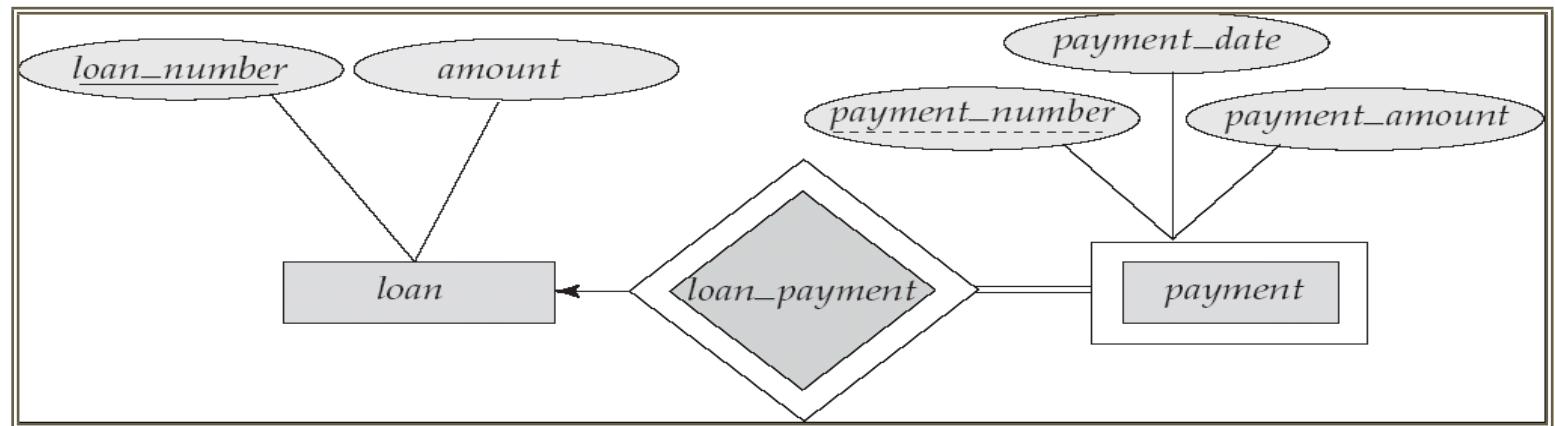


# Varian Entitas

1. Entitas Kuat: Keberadaannya tidak tergantung dari entitas lain
2. Entitas Lemah merupakan Himpunan entitas yang tidak memiliki primary key
  - Keberadaan entitas lemah bergantung pada eksistensinya dalam sebuah relasi terhadap entitas lainnya.
  - Identifikasi relasi dengan menggunakan double diamond
  - Diskriminator (atau key parsial) dari himpunan entitas lemah adalah atribut-atribut yang dapat membedakan entitas-entitas yang ada di himpunan entitas lemah.
  - Primary key dari himpunan entitas lemah dibentuk dari primary key himpunan entitas (kuat) dimana entitas lemah bergantung, serta diskriminator dari entitas lemah itu sendiri

## Varian Entitas (Lanj.)

- Penggambaran himpunan entitas lemah menggunakan dobel persegi panjang
- Diskriminator dari himpunan entitas lemah digambarkan menggunakan garis bawah yang terputus-putus
- Payment-number – merupakan contoh diskriminator himpunan entitas payment
- Primary key untuk payment – (loan-number, payment-number)



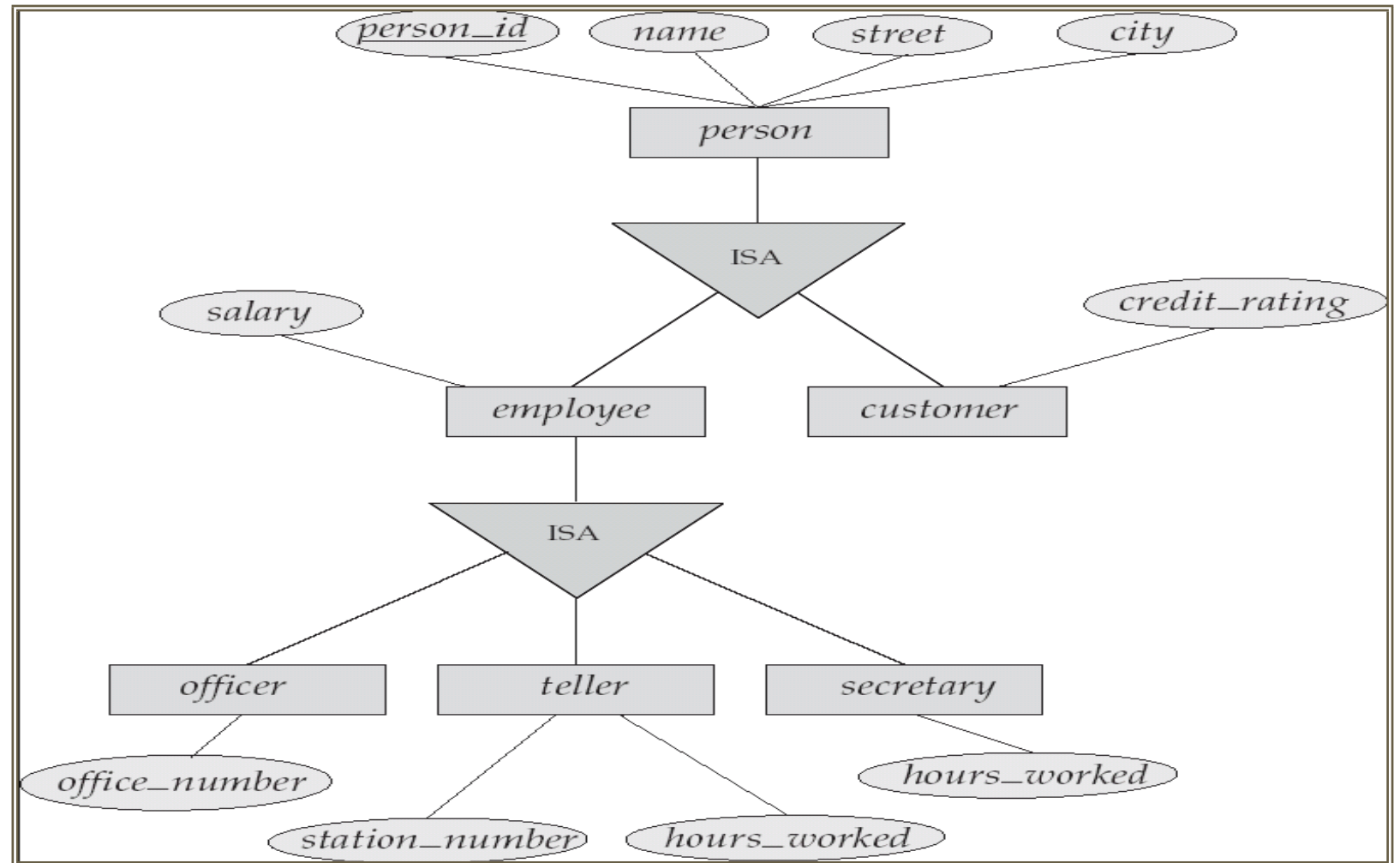
## Varian Entitas (Lanj.)

3. Spesialisasi merupakan proses desain top-down; dengan mendesain subgrouping didalam himpunan entitas yang berbeda dari himpunan entitas lain
  - ❧ Subgrouping ini menjadi himpunan entitas yang levelnya lebih rendah dan memiliki atribut yang tidak dimiliki pada level atasnya.
  - ❧ Digambarkan dengan komponen triangle berlabel ISA (Contoh : customer “is a” person).
  - ❧ Inheritan Atribut– Semua atribut dan relasi pada level lebih tinggi akan diturunkan pada himpunan entitas level bawahnya.
4. Generalisasi merupakan proses desain bottom-up, mengkombinasikan jumlah himpunan entitas yang mempunyai fitur sama ke level yang lebih tinggi

Spesialisasi dan generalisasi merupakan kebalikan satu sama lain

## Varian Entitas (Lanj.)

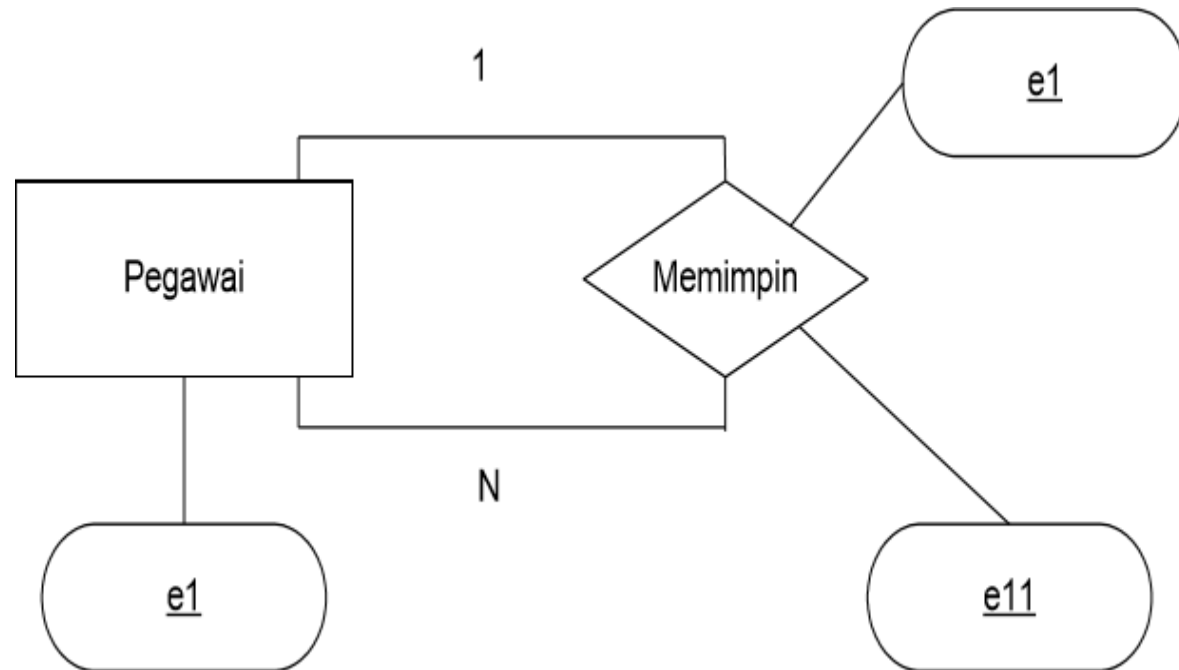
### Contoh Spesialisasi





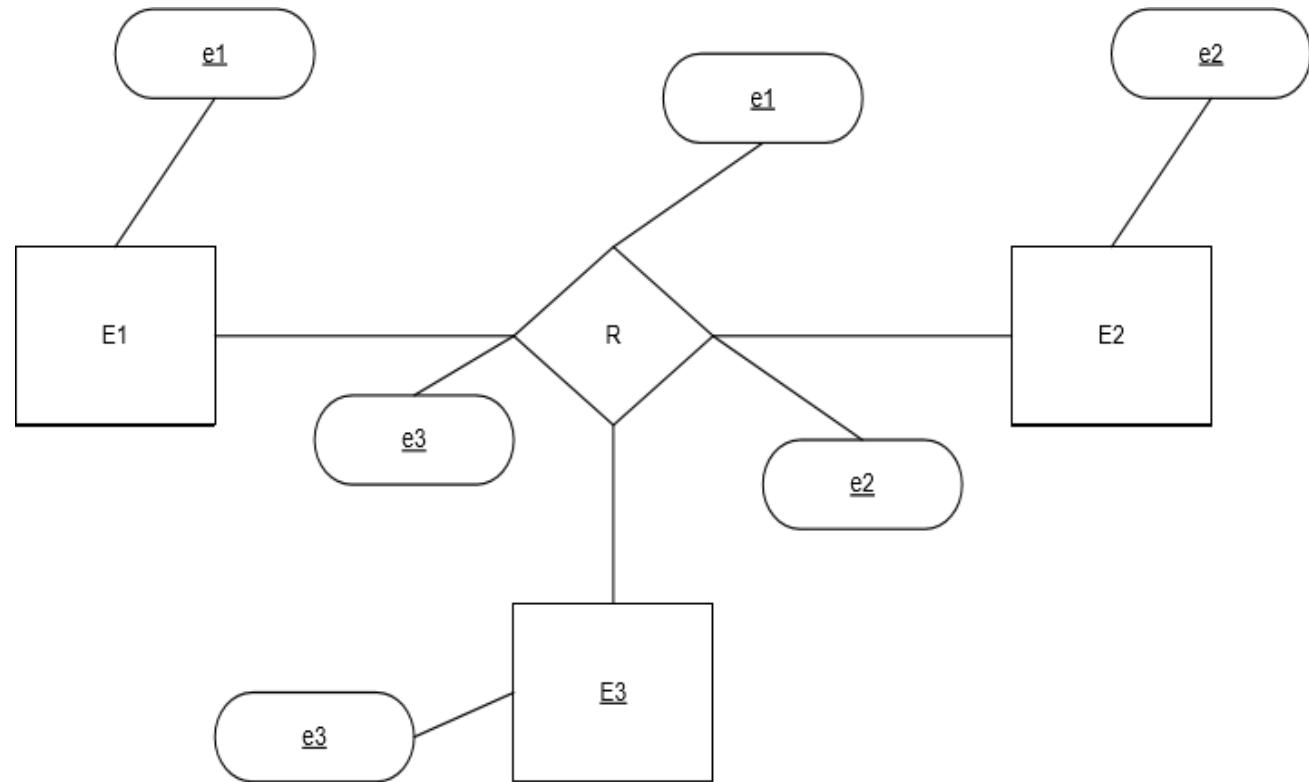
## Varian Relasi

Relasi Tunggal merupakan Relasi yang dihasilkan dari 1 entity set.  
Misalnya :



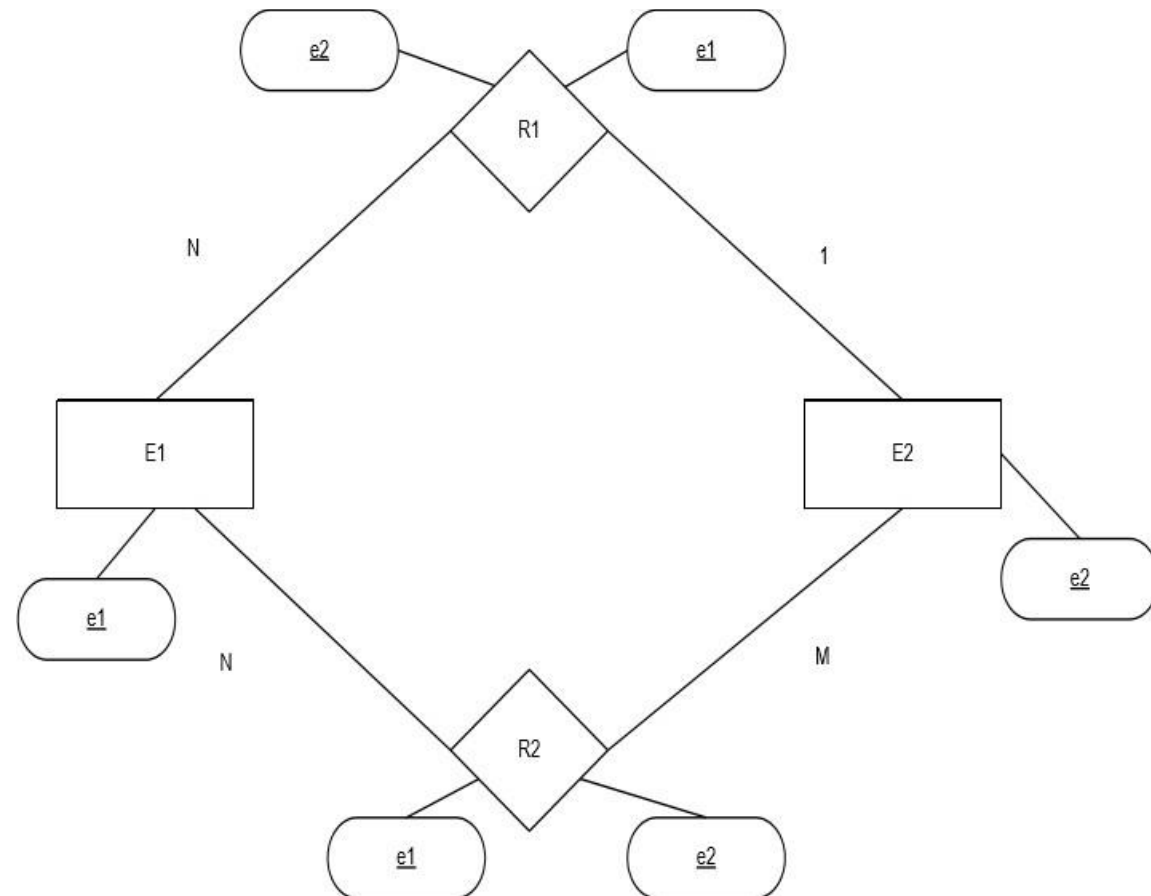
## Varian Relasi (Lanj.)

Relasi Multi entitas merupakan Relasi yang terjadi apabila memang ketiganya saling berelasi

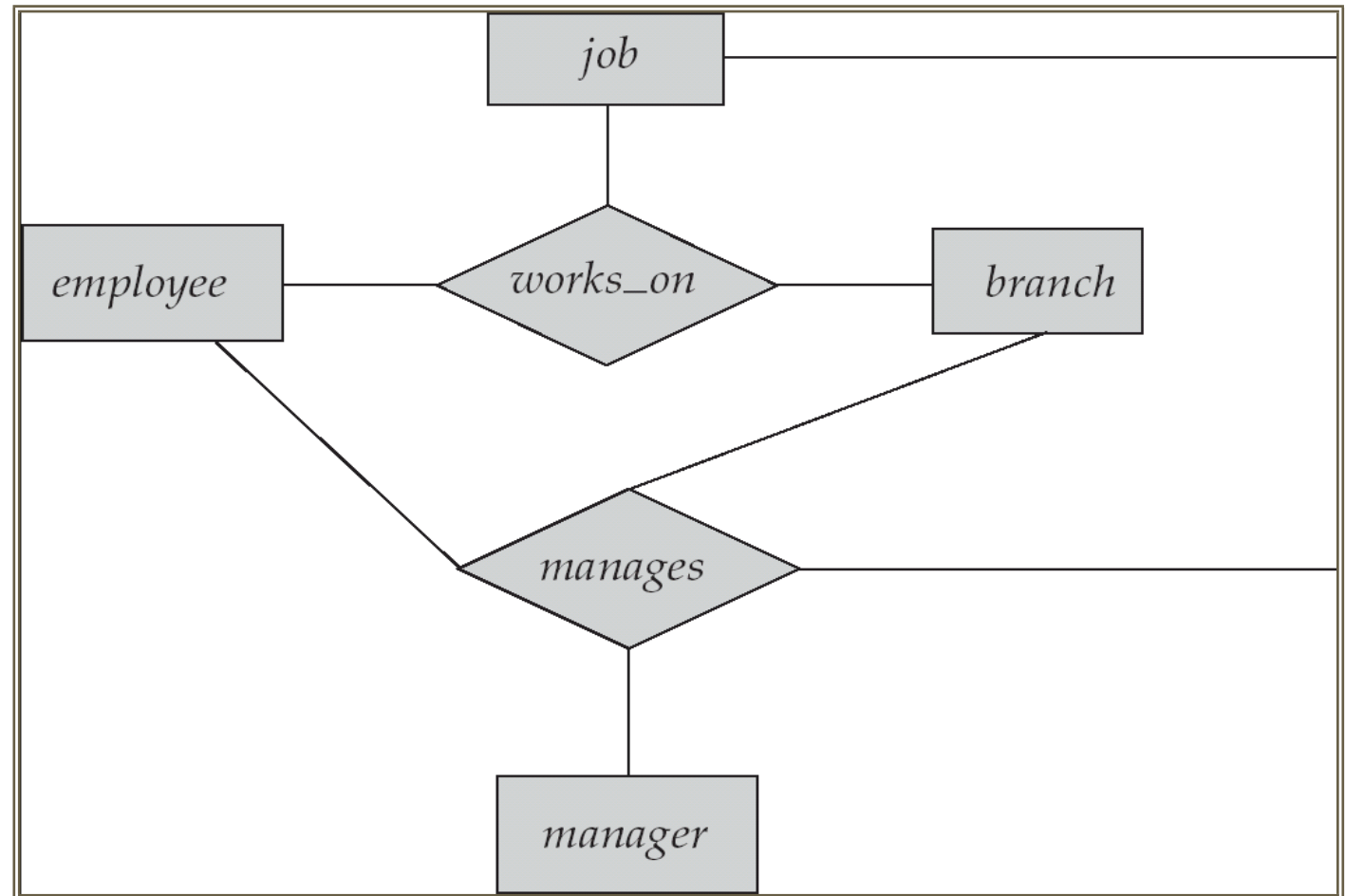


## Varian Relasi (Lanj.)

Relasi Ganda merupakan dua entitas yang memiliki hubungan / relasi lebih dari satu



**Agregasi → Pada contoh ternary relationship works-on, misalkan kita ingin menyimpan informasi manager untuk pekerjaan yang dilakukan oleh seorang employee pada sebuah branch**

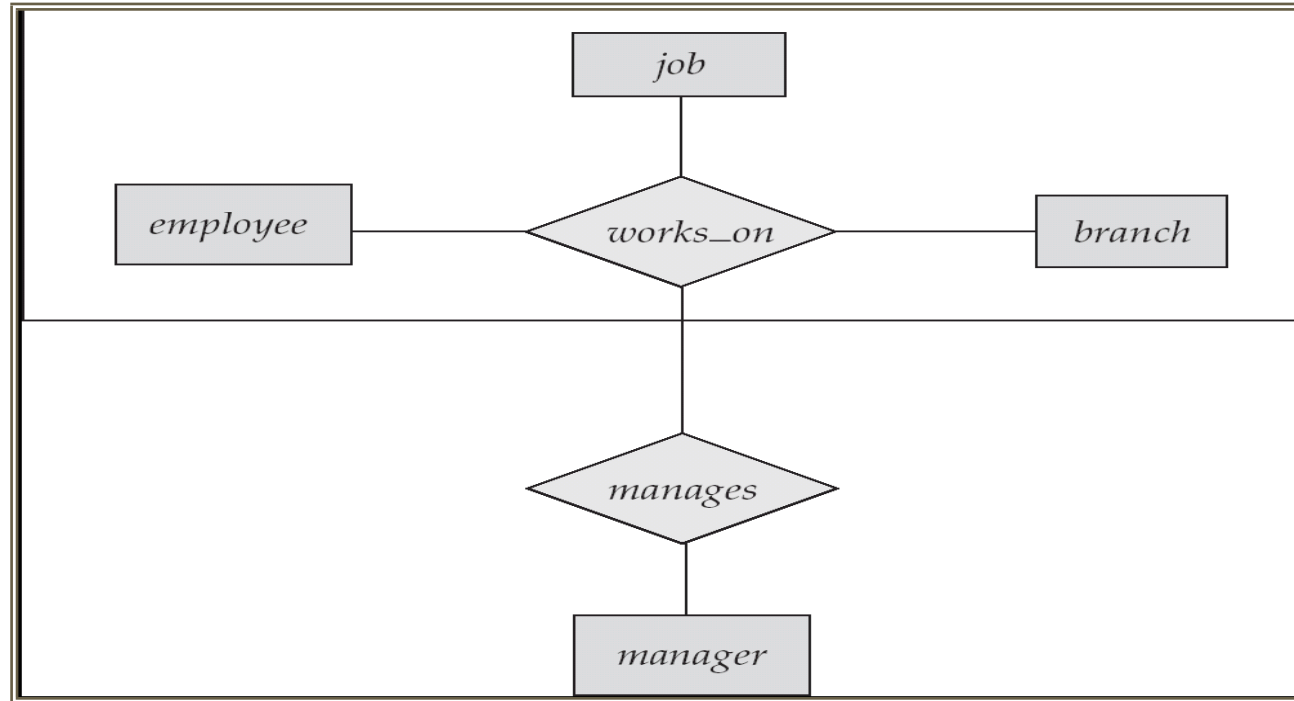


## Varian Relasi (Lanj.)

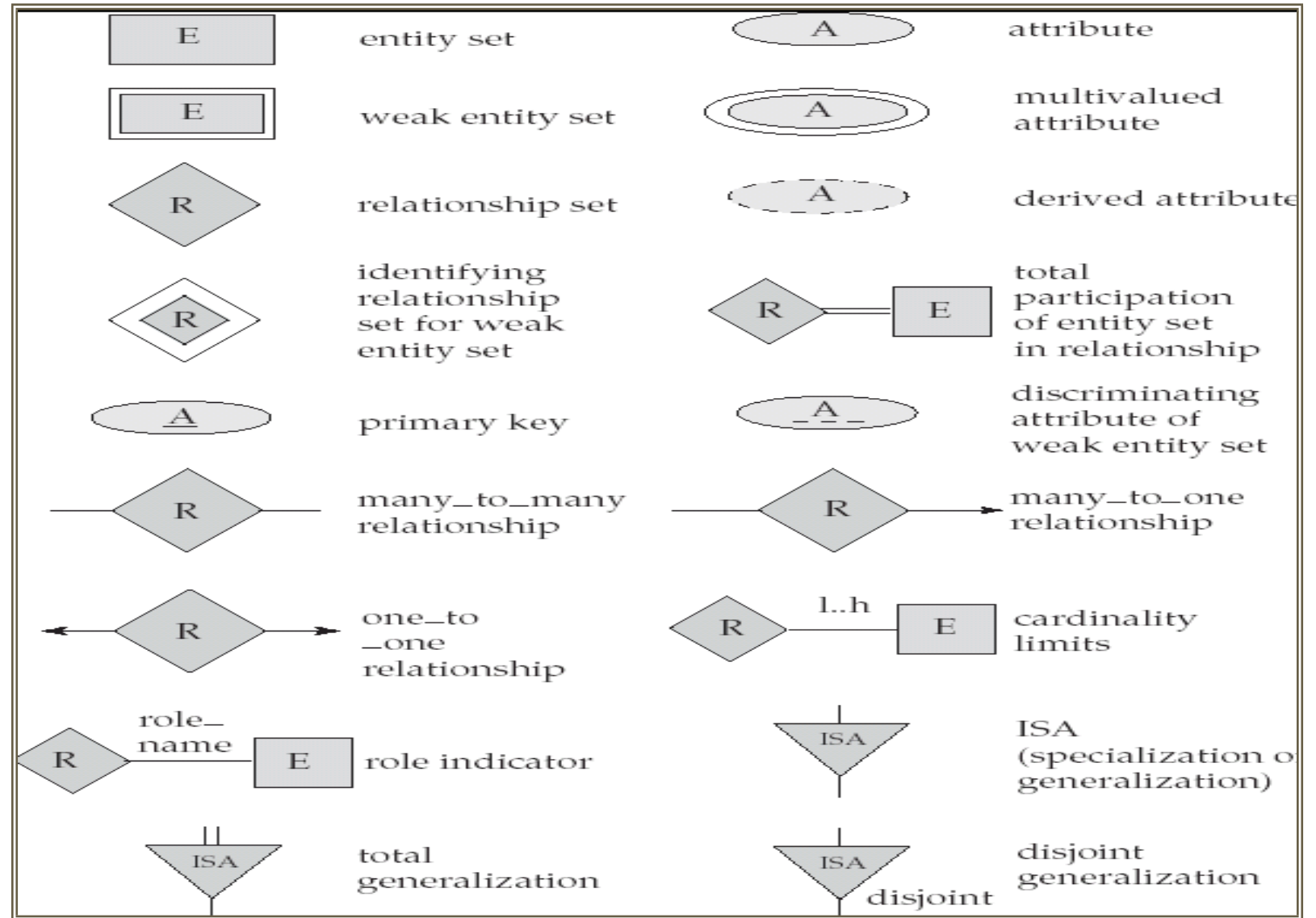
- Relationship sets works-on dan manages menunjukkan informasi yang overlap
- Setiap anggota relationship manages berkaitan dengan sebuah relationship works-on
- Akan tetapi, beberapa anggota relationship works-on mungkin tidak berkaitan dengan relationship manages. Sehingga relationship workson tidak dapat dihilangkan
- Redundansi tersebut dapat diatasi dengan agregasi – Relationship dijadikan sebagai sebuah entitas abstrak – Relationship antar relationship diijinkan – Abstraksi relationship menjadi entitas baru
- Tanpa redundansi, diagram berikut menunjukkan :
  - Seorang employee melakukan sebuah pekerjaan tertentu pada sebuah branch
  - Kombinasi dari seorang employee, branch dan job dapat diasosiasikan dengan seorang manager

## Varian Relasi (Lanj.)

Hasil Agregasi:



# Simbol dalam ER



## Simbol dalam ER (Lanj.)

