

# 4

## Pemodelan Basis Data dengan EER ( PART 1 )

CSF2600700 - BASIS DATA

enhanced





## Acknowledgements

This slide is a modification to supplementary slide of “Database System”, 6th edition, Elmasri/Navathe, 2011: **Chapter 4** used in “Basis Data” course in academic years 2018/2019 in the Faculty of Computer Science, Universitas Indonesia.

Additional resources: presentation prepared by Prof Steven A. Demurjian, Sr  
(<http://www.engr.uconn.edu/~steve/courses.html>)

## Tujuan Pembelajaran

Setelah mengikuti pembelajaran pada topik ini, jika diberikan requirement basis data, Anda diharapkan dapat **memodelkan** basis data dengan tepat menggunakan **Enhanced Entity Relationship Diagram**



# Outline

## 1. Latar Belakang

## 2. Superclass/Subclass Relationship

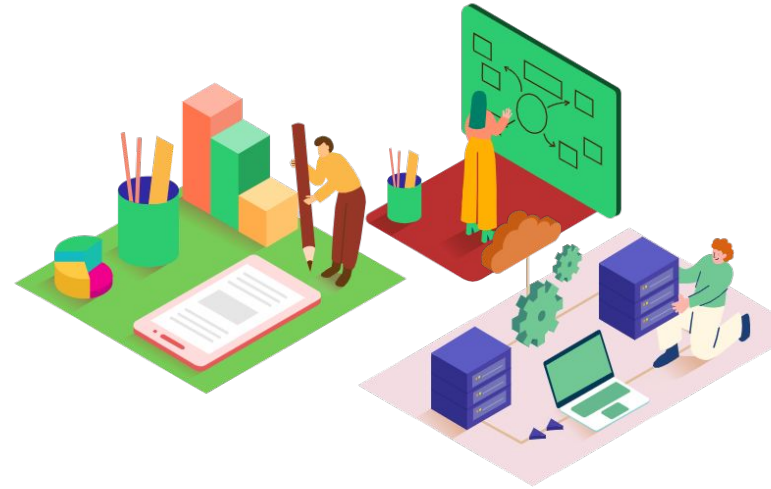
## 3. Spesialisasi dan Generalisasi

## 4. Hierarchy dan Lattice

## 5. Pemodelan dengan Categories

## 6. Higher Degree Relationship

## 7. Kapan Kita Menggunakan EER?



# Mengapa Perlu Enhanced ER?

ER cukup digunakan untuk memodelkan skema basis data 'tradisional' (aplikasi pemrosesan data pada bisnis dan industri pada umumnya)

Sejak akhir tahun 70-an, dirasakan perlu untuk merancang skema yang dapat merepresentasikan sifat-sifat dan batasan-batasan data dengan lebih tepat, terutama untuk aplikasi-aplikasi baru di berbagai bidang (CAD, CAM, GIS, dll)

Hal ini memacu perkembangan konsep-konsep semantic data modeling yang ditambahkan ke model ER yang telah ada



## Konsep-Konsep Model EER

Model Enhanced/  
Extended ER

=

Semua Konsep  
tentang ER

+

Konsep Subclass/Superclass,  
Specialization/Generalization,  
Categories, Attribute Inheritance

- 
- Model EER digunakan untuk merepresentasikan aplikasi dengan lebih **lengkap** dan **lebih akurat**, jika diperlukan
  - Model EER mengandung beberapa konsep **object oriented**, misal: inheritance

# Outline

1. Latar Belakang

2. Superclass/Subclass Relationship

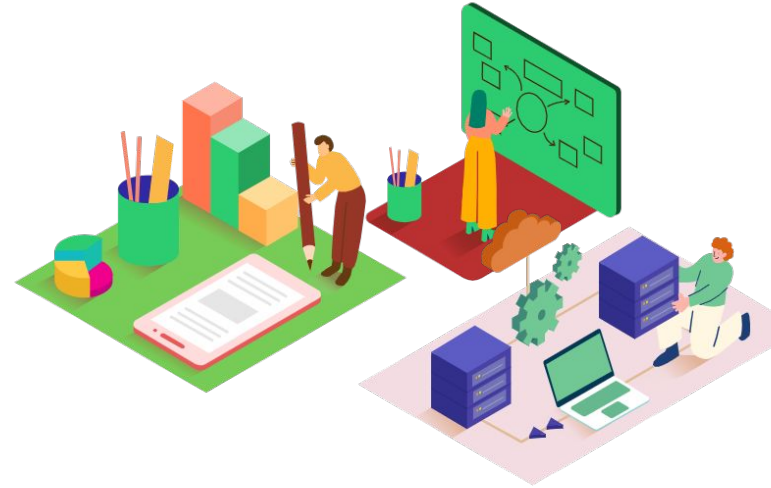
3. Spesialisasi dan Generalisasi

4. Hierarchy dan Lattice

5. Pemodelan dengan Categories

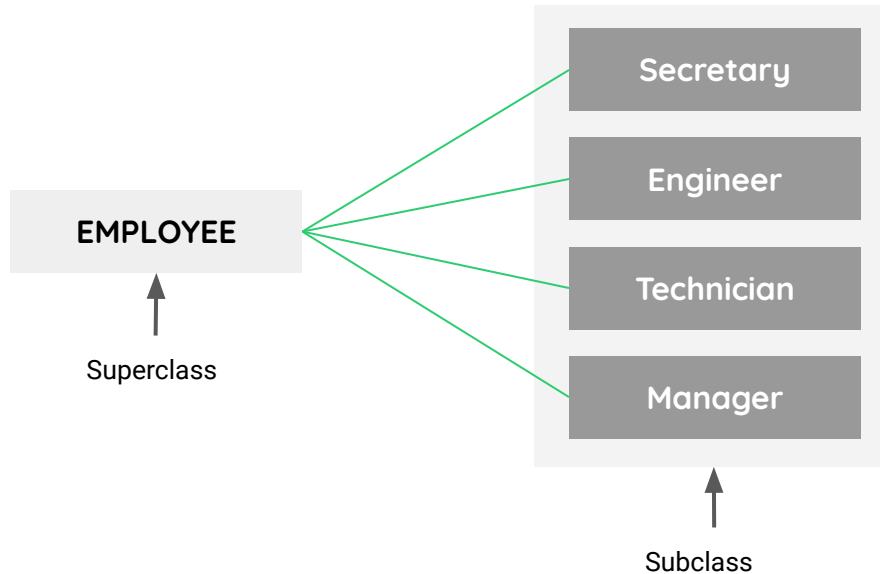
6. Higher Degree Relationship

7. Kapan Kita Menggunakan EER?



# Subclass dan Superclass

Misal EMPLOYEE dapat dikategorikan menjadi 4 kelompok



Subclass merepresentasikan **entity yang sama dengan superclass**, namun memiliki peran spesifik tertentu.

Entity dalam subclass merupakan **anggota superclass**, namun tidak sebaliknya



# Superclass/Subclass Relationship

Superclass/Subclass Relationship adalah relationship antara sebuah superclass dengan salah satu subclass-nya.

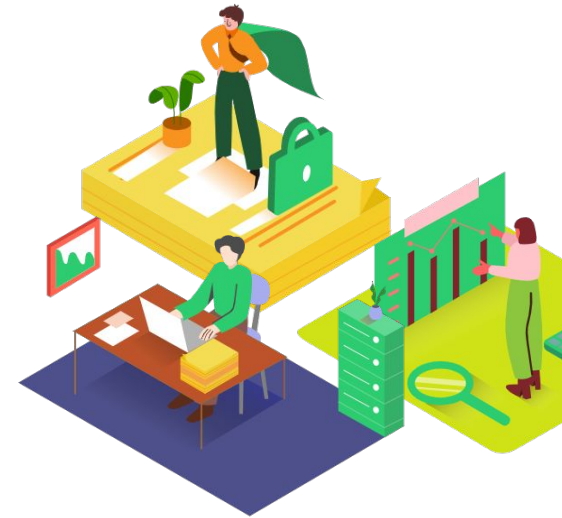
## Contoh:

- Employee/Secretary
- Employee/Technician

Disebut juga dengan IS-A relationship

## Contoh:

- SECRETARY IS AN EMPLOYEE
- TECHNICIAN IS AN EMPLOYEE



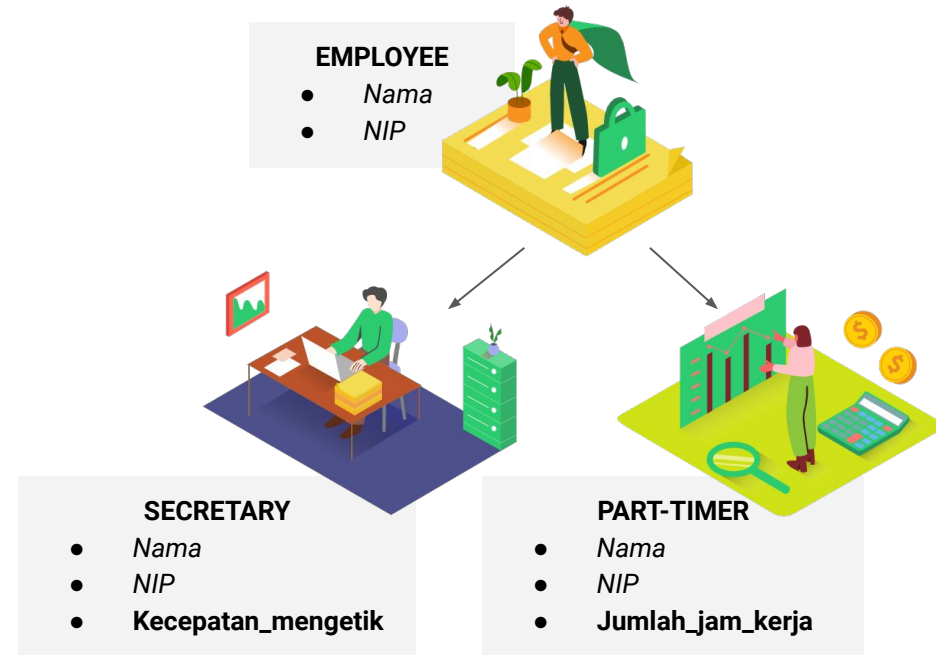
# Type Inheritance

Suatu entity yang merupakan anggota sebuah subclass mewarisi (inherits)

**Contoh:**

- semua **attribute**, dan
- semua **relationship**

dari entity yang merupakan anggota superclass.



# Outline

1. Latar Belakang

2. Superclass/Subclass Relationship

3. Spesialisasi dan Generalisasi

4. Hierarchy dan Lattice

5. Pemodelan dengan Categories

6. Higher Degree Relationship

7. Kapan Kita Menggunakan EER?



# Spesialisasi

Spesialisasi adalah proses mendefinisikan himpunan subclass-subclass dari sebuah entity type (superclass)

Dilakukan berdasarkan karakteristik tertentu yang dapat membedakan entity pada superclass

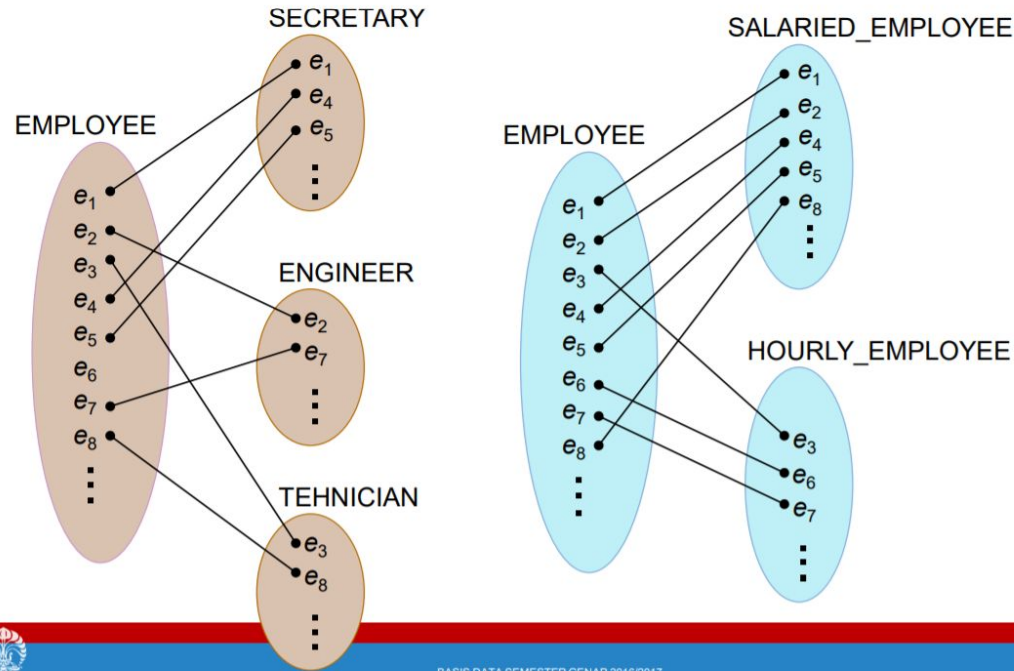
Suatu superclass dapat memiliki beberapa spesialisasi berdasarkan karakteristik yang berbeda

## Contoh:

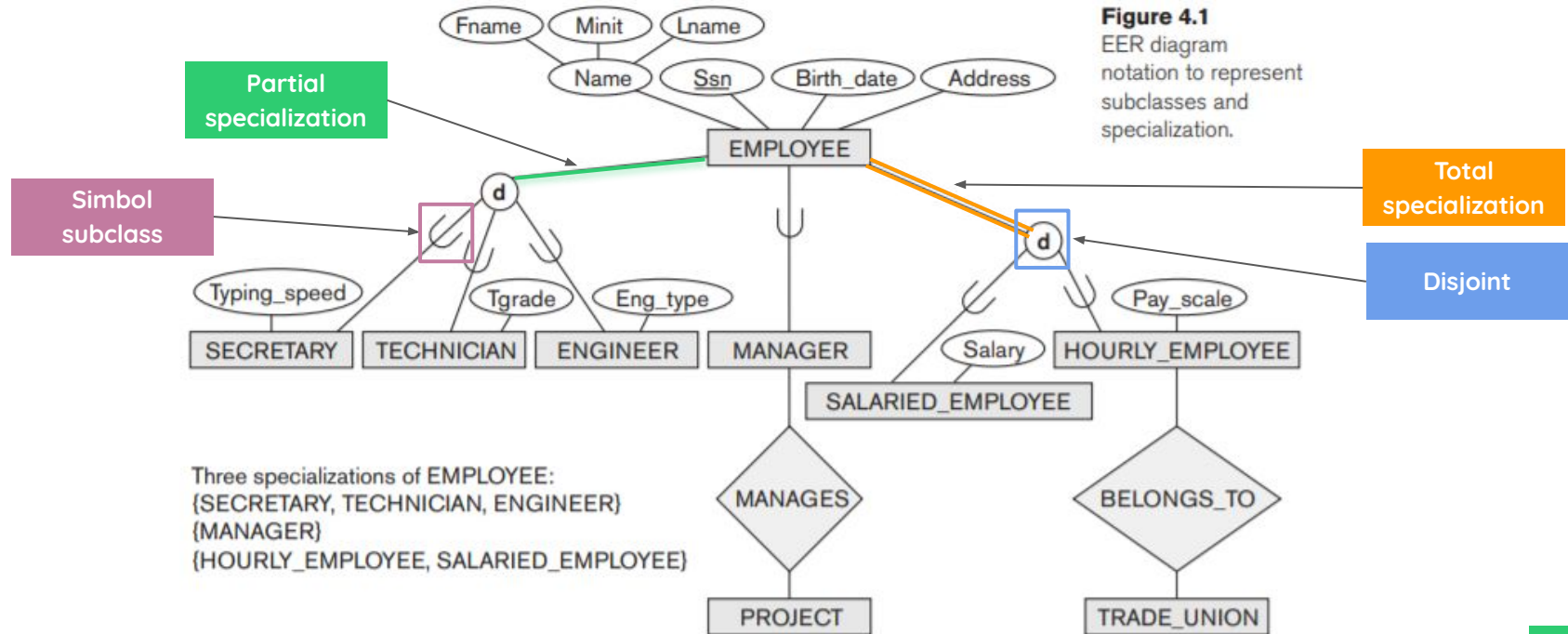
- SECRETARY, ENGINEERS, TECHNICIAN adalah spesialisasi dari EMPLOYEE berdasarkan attribute job\_type
- SALARIED\_EMPLOYEE dan HOURLY\_EMPLOYEE adalah spesialisasi dari EMPLOYEE berdasarkan metode pembayarannya.



## Contoh Spesialisasi



## Notasi Spesialisasi dalam EER

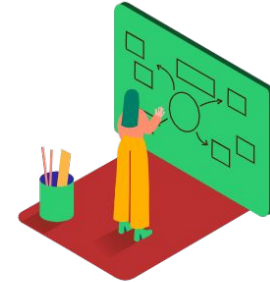


## Manfaat Spesialisasi

Mendefinisikan **himpunan subclass-subclass** dari suatu entity type

Menggambarkan **attribute** spesifik untuk tiap subclass

Menggambarkan relationship spesifik **antara suatu subclass dengan entity type lain** atau dengan subclass lain



# Generalisasi

## Kebalikan dari proses spesialisasi

Dilakukan dengan mengidentifikasi attribute-attribute yang sama dan melakukan generalisasi ke sebuah superclass

### Contoh:

→ TRUCK & CAR dapat digeneralisasi menjadi VEHICLE

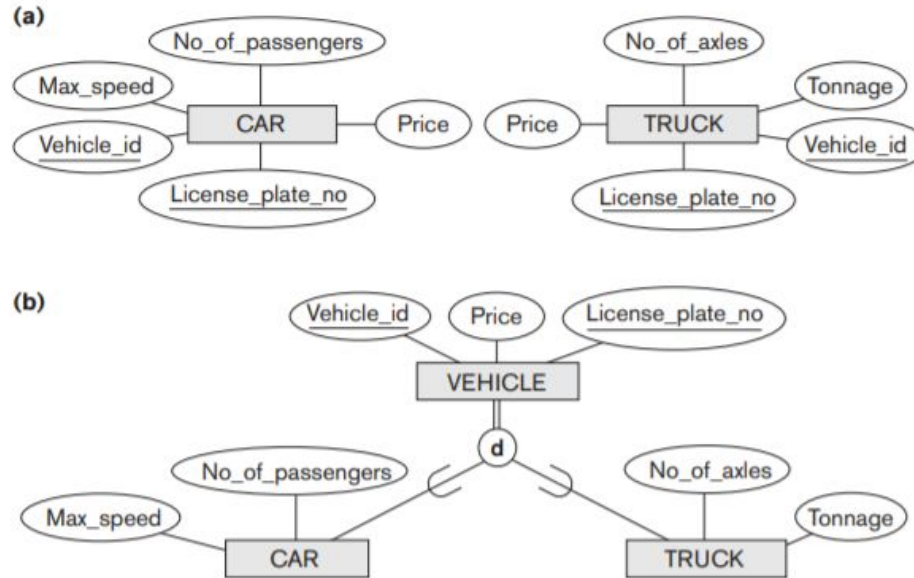




## Contoh Generalisasi

**Figure 4.3**

Generalization. (a) Two entity types, CAR and TRUCK.  
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.



## Generalisasi vs Spesialisasi

Kadang-kadang notasi spesialisasi dan generalisasi dibedakan:

- Arah panah menuju superclass menunjukkan generalisasi
- Arah panah menuju subclass menunjukkan spesialisasi

Di sini kita tidak membedakan notasi dengan arah panah, karena seringkali subyektif sesuai dengan proses yang dilakukan pada suatu situasi tertentu

## Constraints untuk Spesialisasi dan Generalisasi (1)

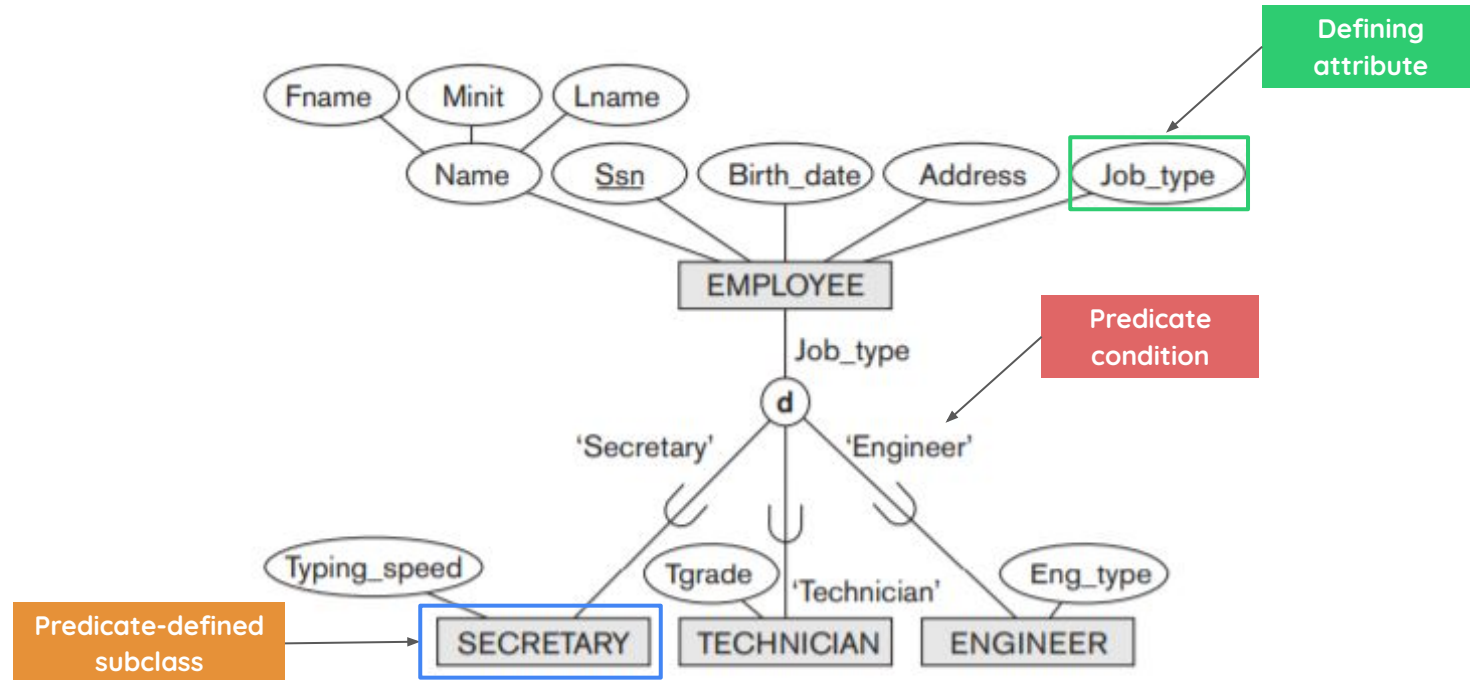
Spesialisasi berdasarkan attribute

- Spesialisasi dilakukan berdasarkan attribute dari superclass (defining attribute)
- Contoh: job\_type

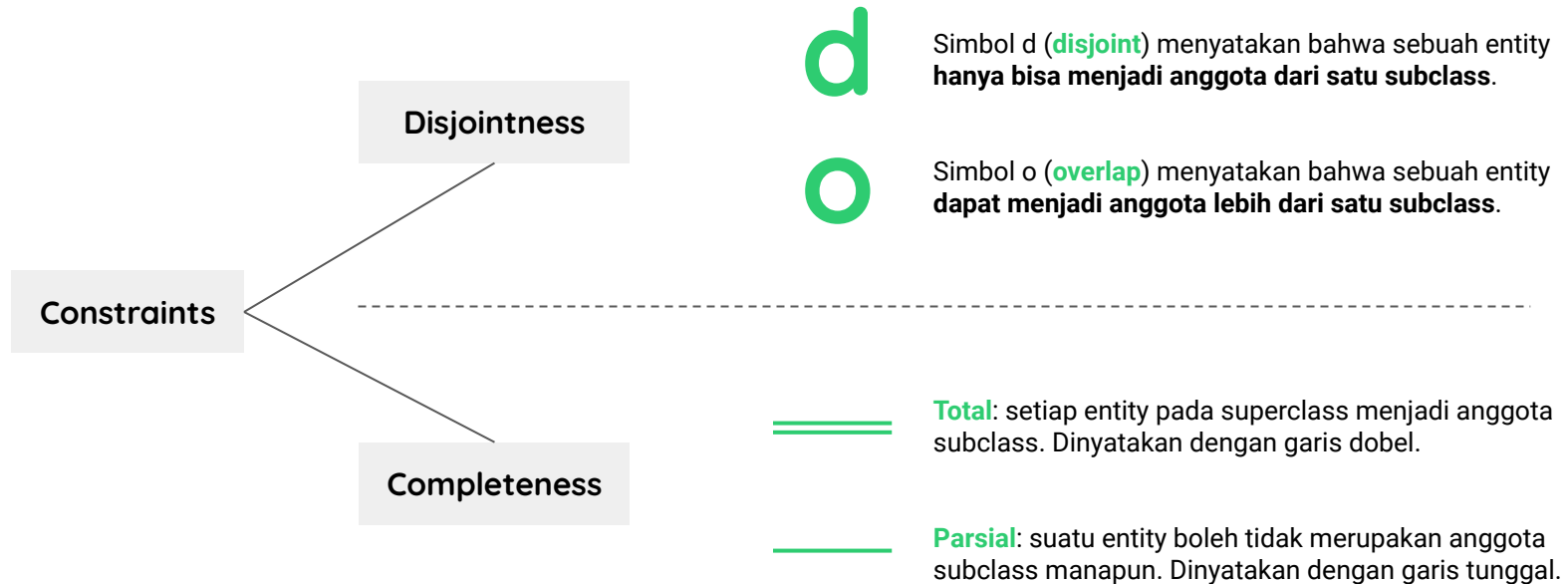
Subclass yang ditentukan pengguna

- Keanggotaan entity dalam suatu subclass ditentukan oleh pengguna

## Constraints untuk Spesialisasi dan Generalisasi (2)



## Constraints untuk Spesialisasi dan Generalisasi (3)



## Constraints untuk Spesialisasi dan Generalisasi (4)

Dari constraints tersebut, ada **4 macam bentuk spesialisasi/generalisasi**

Disjoint, **total**

Overlap, **total**

Disjoint, **parsial**

Overlap, **parsial**

Generalisasi umumnya bersifat **total** karena superclass diturunkan dari subclass-subclassnya.

## Constraints untuk Spesialisasi dan Generalisasi (5)

### Contoh Kasus 1

#### → Disjoint partial

Pada basis data sebuah perusahaan, data EMPLOYEE yang memiliki subclass SECRETARY, TECHNICIAN, dan ENGINEER. Seorang pegawai **boleh merupakan SECRETARY atau TECHINICIAN atau ENGINEER atau tidak ketiganya** (misalnya ACCOUNTANT), namun **tidak boleh memiliki pekerjaan lebih dari satu**, misalnya SECRETARY sekaligus TECHNICIAN atau TECHNICIAN sekaligus ENGINEER.

## Constraints untuk Spesialisasi dan Generalisasi (6)

### Contoh Kasus 2

#### → Disjoint total

Merujuk pada contoh kasus sebelumnya, namun dalam hal ini seorang PEGAWAI **HARUS memiliki SALAH SATU pekerjaan** diantara SECRETARY, TECHNICIAN, atau ENGINEER. Jadi dalam kasus ini **TIDAK ADA PEGAWAI yang memiliki pekerjaan selain ketiga pekerjaan tersebut** (tidak ada pekerjaan ACCOUNTANT lagi)



## Constraints untuk Spesialisasi dan Generalisasi (7)

### Contoh Kasus 3 & 4

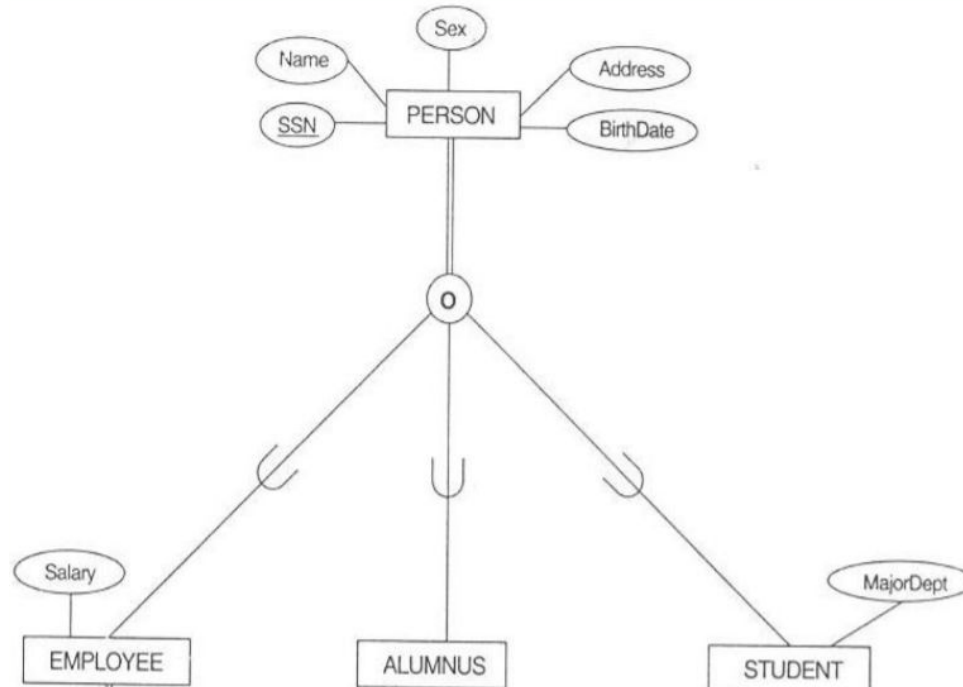
#### → Overlap partial

Pada kasus ini, seorang EMPLOYEE **dapat memiliki satu atau lebih jabatan**, misalnya seseorang dapat bekerja sebagai TECHNICIAN sekaligus ENGINEER. Namun **bisa pula seorang EMPLOYEE bukan merupakan anggota dari subclass manapun**, misalnya ia merupakan seorang MANAGER

#### → Overlap total

Pada kasus ini, seorang EMPLOYEE **dapat terdaftar sebagai salah satu (MINIMAL)**, dua, atau ketiga pekerjaan antara SECRETARY, TECHNICIAN, dan ENGINEER.

## Contoh Spesialisasi Overlap Total



**Next:**  
**Pemodelan Basis Data dengan EER**  
Bagian 2

