

# Perancangan Basis Data

## Pertemuan 9

Class Diagram & ORM

# Tujuan Pertemuan

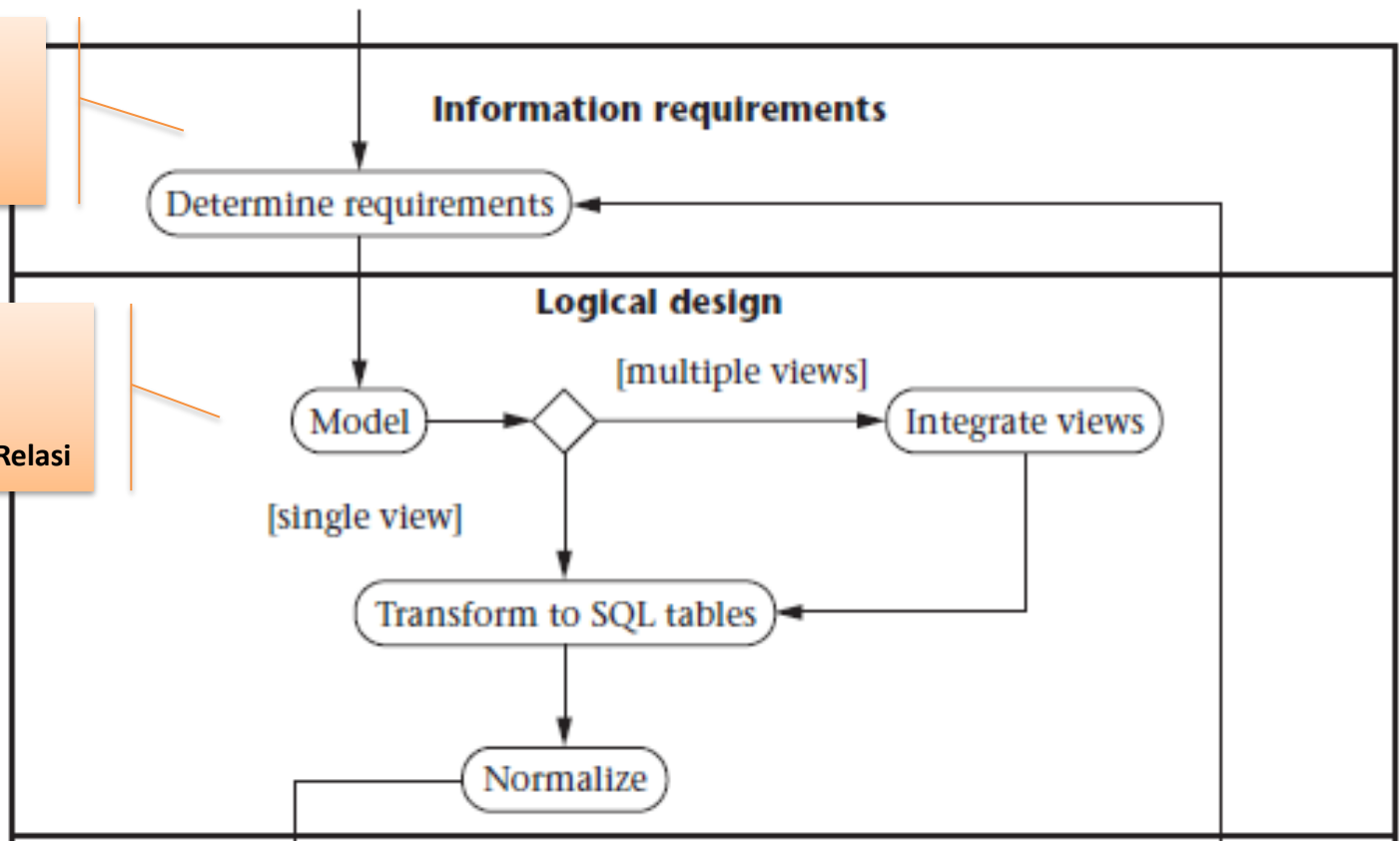
- Mahasiswa mampu melakukan disain detailed specification model structure menggunakan diagram ORM (Object Role Modeling).
- Mahasiswa mampu melakukan pemetaan (mapping) dari disain konseptual (ORM) ke disain logis (diagram Class)
- Mahasiswa mampu melakukan disain detailed specification model structure menggunakan diagram class.

**Behaviour Diagram**

Contoh:  
OOAD: Use Case  
SSAD: DFD

**Structure Diagram**

Contoh:  
OOAD: class & ORM  
SSAD: ERD / Normalisasi Relasi



# Fase Logical Design

- Pada database life cycle setelah melalui fase information requirements (*pada UML menggunakan Use Case*), selanjutnya memasuki fase logical design.
- Pemodelan informasi untuk database melibatkan pemodelan data, pada UML **pemodelan data menggunakan structure diagram**, yaitu;
  - Diagram Class → untuk skema data
  - Diagram Object (ORM) → untuk populasi data

# Pemodelan Data (Data Modeling)

- Cara terbaik untuk **membuat model data dengan UML** adalah;
  - Membuat model ORM
  - Memetakan model ORM ke UML class diagrams
- Object – Role Modeling (ORM) digunakan untuk klarifikasi konsep pemodelan data dalam UML.
- ORM hanya fokus pada pemodelan data

# ORM?

- **Object-Role Modeling (ORM)** adalah suatu metode untuk pemodelan dan pengolahan (querying) suatu system informasi pada level konseptual.
- Method **ORM** = method **NIAM** di eropa (NIAM =Natural language Information Analysis Method).
- ORM menyertakan prosedur-prosedur antara pemetaan level logical dan level konseptual.

# ORM

- ORM menggambarkan kejadian dunia nyata dari sisi object (entitas atau nilai) yang memiliki peran (sebagai bagian dalam relationship).
- Contoh:
  - Anda sekarang sebagai entitas berperan membaca, dan
  - Slide ini berperan sebagai entitas yang dibaca.



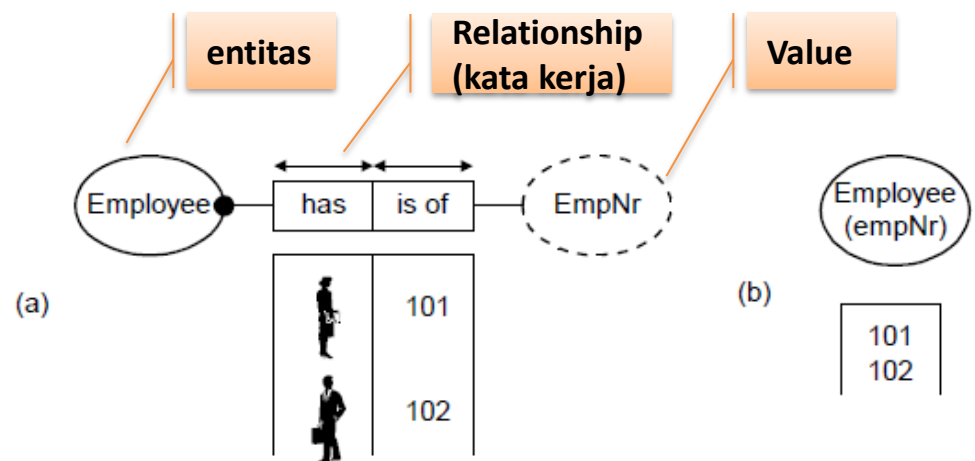
# ORM

- ORM mengklasifikasikan object menjadi;
  - **Entitas** (entity)
  - **Nilai** (values)
- Setiap entitas diidentifikasi oleh **skema referensi** seperti yang digunakan manusia untuk menjelaskan tentang entitas.

Penjelasan verbal gambar:  
Employee **has** EmpNr, atau  
EmpNr **is of** Employee

(a) Contoh model ORM dengan notasi reference.

(b) Reference dapat disingkat dengan menuliskannya di dalam (..) contoh: (empNr)

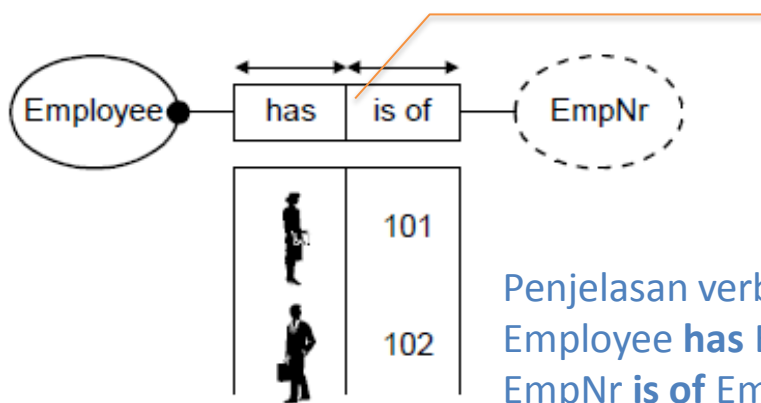




# ORM

## Skema Simple referensi ORM

- **Object** digambarkan dengan **elips**;
  - Object tipe **entitas** → elips dengan **solid line** (garis solid)
  - Object tipe **value** → elips dengan **dashed line** (garis putus)
- **Relationship (reference)** digambarkan dengan kotak, setiap kotak berisi role (aturan) yang menjelaskan hubungan antar object yang terhubung.



Banyaknya role pada relationship disebut **arity**;  
Tingkatan arity; 1 = unary, 2 = binary, 3 = ternary,  
4 = quaternary, 5 quinary.  
(pada prakteknya 80% arity adalah binary)

Penjelasan verbal gambar:  
Employee **has** EmpNr, atau  
EmpNr **is of** Employee

# ORM

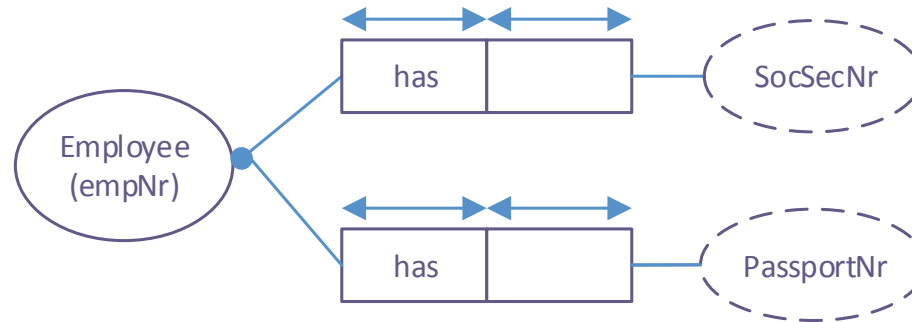
## Notasi Relationship



- ●– (titik) : **mandatory role** (role wajib).  
Artinya: *“Setiap Employee wajib memiliki empNr, minimal satu empNr”*
- ↔ : **Uniqueness**
  - ↔ di kiri  
Artinya: *“Employee hanya boleh memiliki maksimal satu empNr”*
  - ↔ di kanan  
Artinya: *“empNr hanya boleh merujuk kepada maksimal satu Employee”*
- Dengan demikian artinya;  
*“Employee minimal memiliki 1 empNr dan Employee maksimal memiliki 1 empNr dan empNr hanya merujuk kepada maksimal 1 Employee”*
- Berarti empNr merupakan **primary reference** bagi Employee. Untuk itu empNr ditandai dengan (P) pada class diagram.

# ORM

## Notasi Relationship (disjunctive mandatory role)



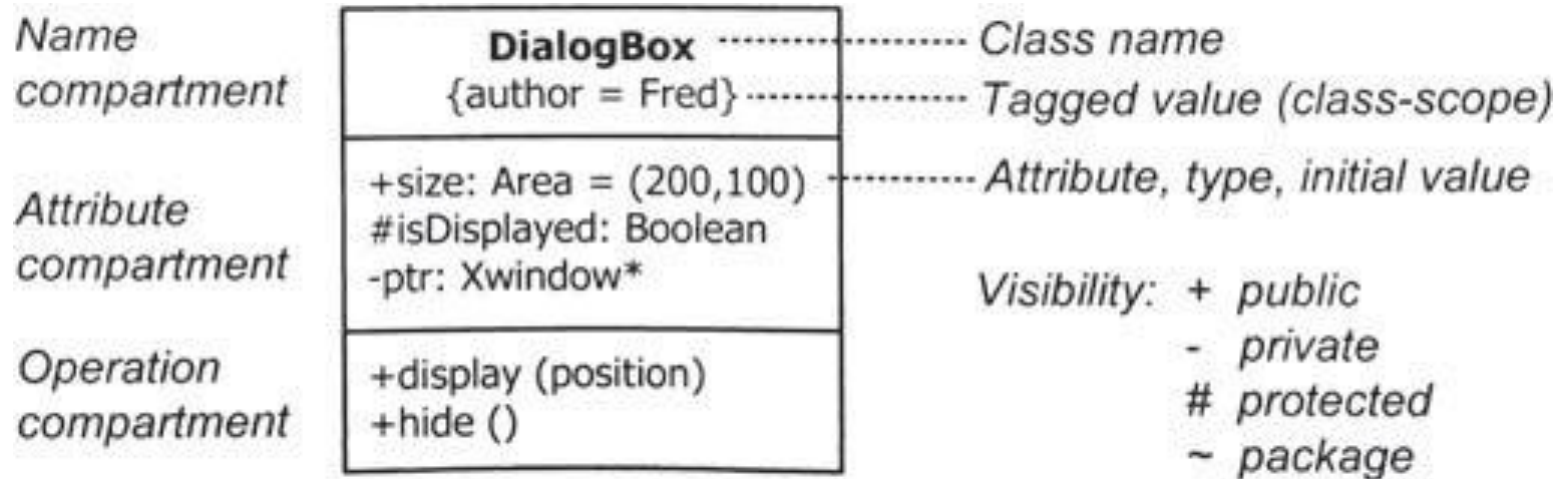
- ●< (titik dengan konektor bercabang) : disjunctive mandatory role (wajib opsi).
- Disjunctive mandatory role merupakan role wajib dengan opsi, minimal satu dari role pada titik percabangan tersebut harus ada.

*Artinya: "Setiap Employee wajib memiliki minimal satu SocSecNr **atau** satu PassportNr atau kedua-duanya (satu SocSecNr dan satu PassportNr)".*

# Pemodelan Data (Data Modeling)

## Notasi Diagram Class

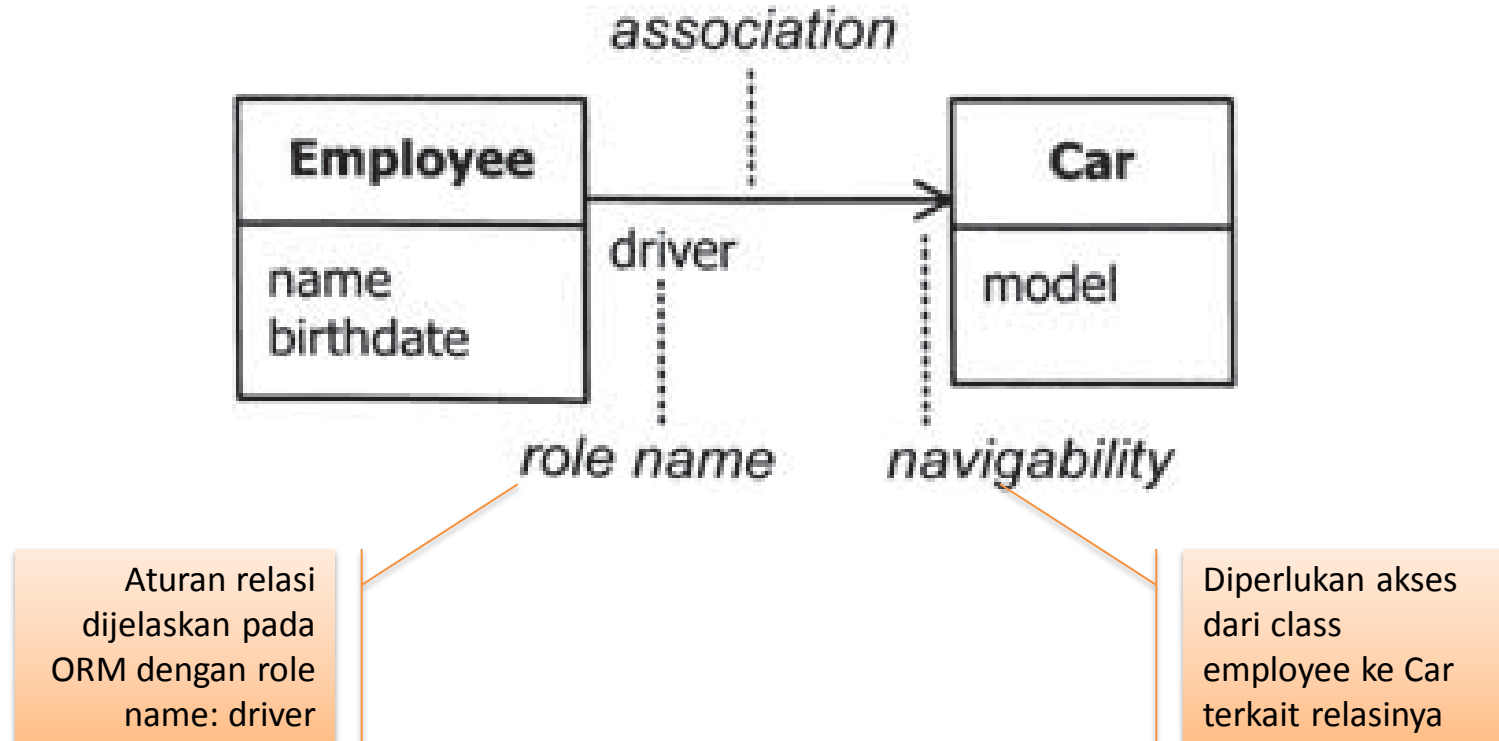
- **Pembuatan Class:**



# Pemodelan Data (Data Modeling)

## Notasi Relasi antar Class

- Pembuatan Class:



# Notasi Diagram Class untuk Analisis Konseptual

- Untuk penggunaan analisis konseptual, **navigability** dan **visibility attribute** dihilangkan.

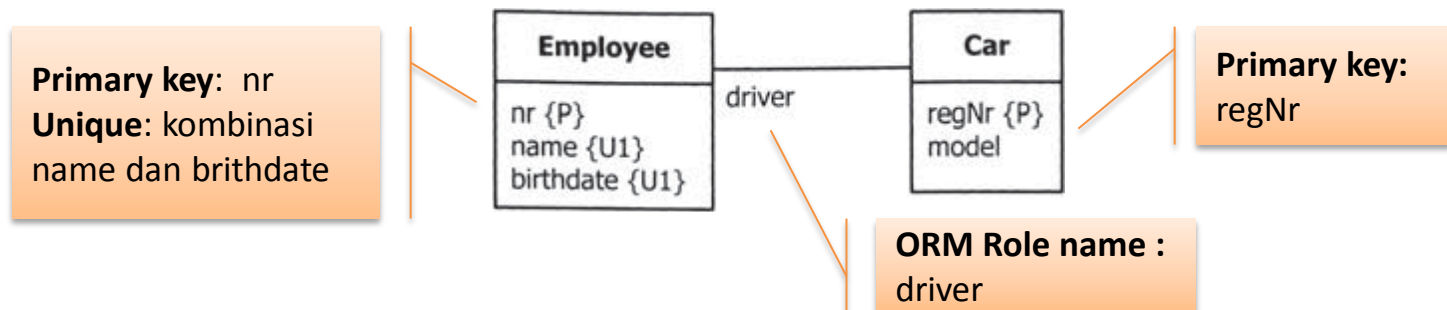
*Dengan demikian sekarang diagram class mirip dengan model ER.*

- Pada UML tidak ada standar penulisan yang menunjukkan bahwa suatu attribute adalah **attribute identitas (primary key)**.

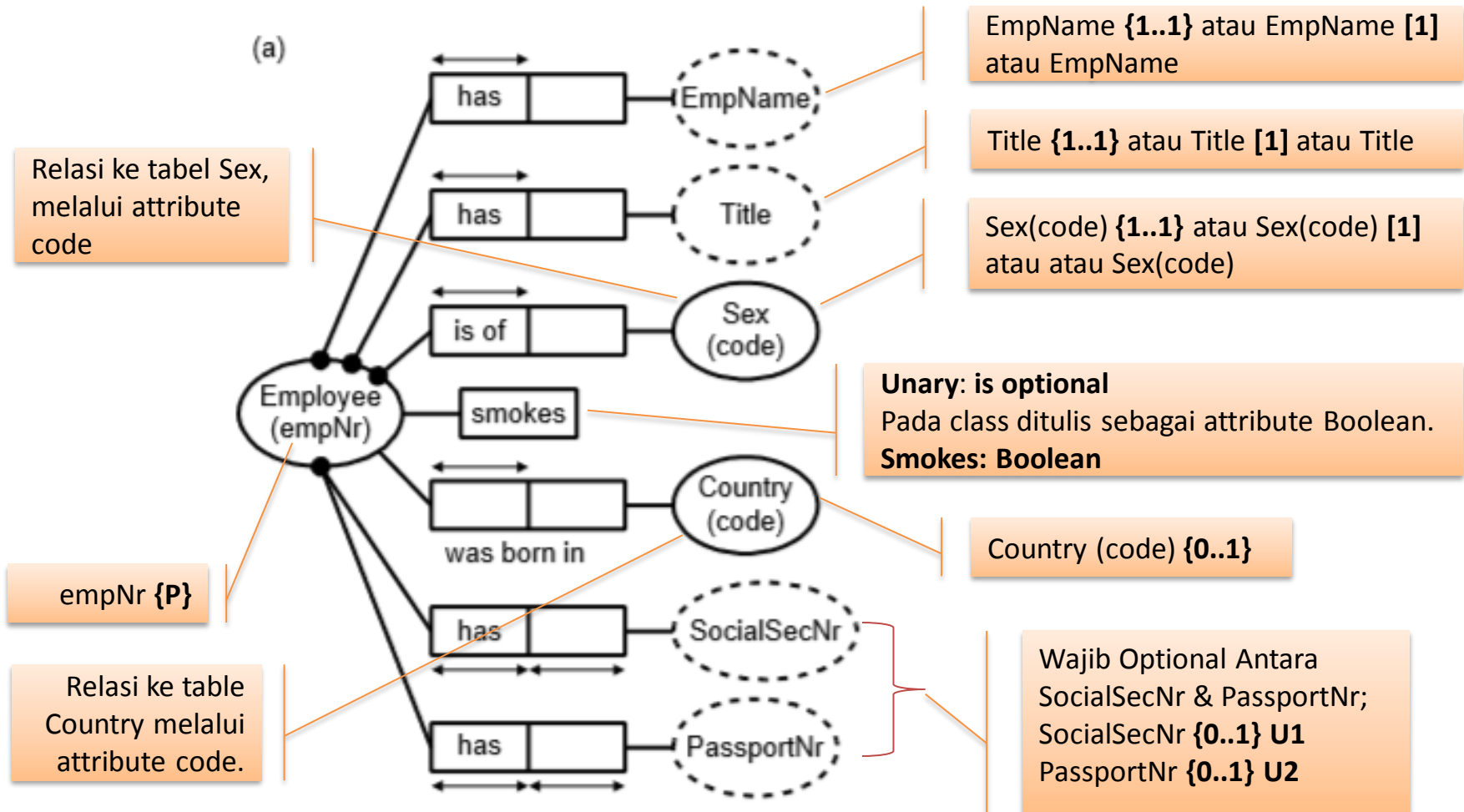
- **Menambahkan identitas attribute;**

- {P} untuk primary key
- {Un} untuk unique

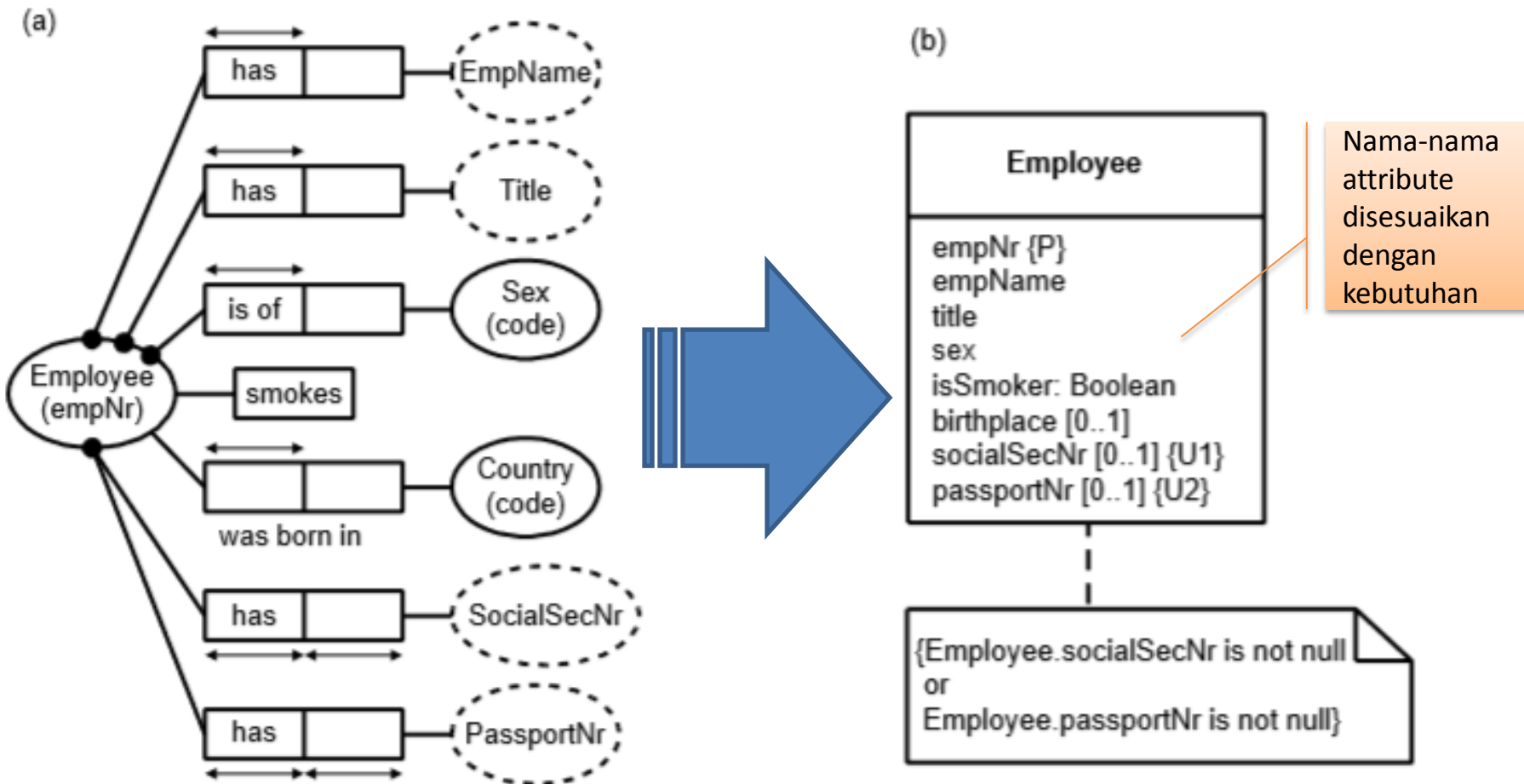
*n : (n>0) digunakan untuk menghindari kerancuan jika unique digunakan pada kombinasi attribute.*



# Pemetaan ORM ke UML Class Diagram

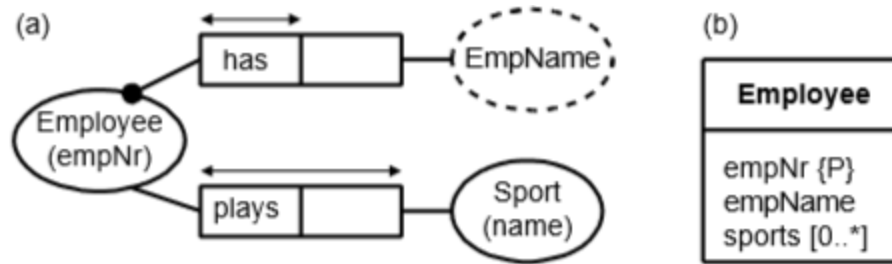


# Pemetaan ORM ke UML Class Diagram





# Pemetaan ORM ke UML Class Diagram Multi-valued Attributes



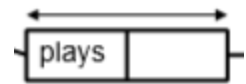
Perhatikan skema referensi Employee(empNr) plays Sport(name);

- Sports [0..\*] pada class: **multi-valued attribute**

- **0** : tidak ada **titik** pd entitas employee

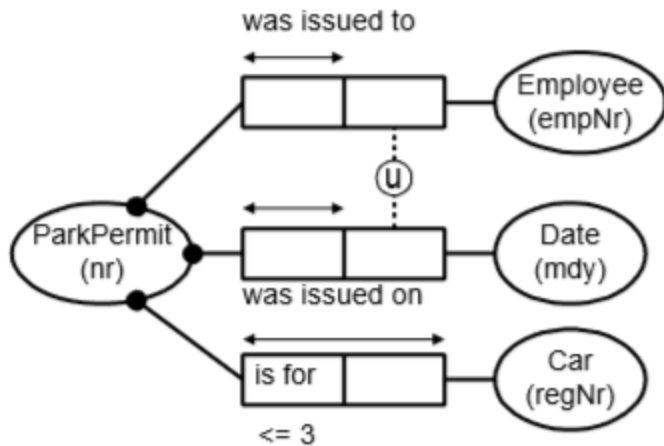
Artinya: *“Employee dimungkinkan tidak memiliki sport”*.

- **\*** :  $\leftrightarrow$  melingkupi 2 role, **uniqueness dari kombinasi 2 role**.



Artinya: *“Employee boleh memiliki lebih dari 1 Sport, dan Sport boleh dimiliki oleh lebih dari 1 Employee, tetapi tidak boleh ada kombinasi Employee & Sport yang sama”*.

- **\*** :  $\leftrightarrow$  melingkupi 2 role, **relationship type many-to-many (m:n)**



- U dalam lingkaran: External unique,  
Artinya:  
“kombinasi Employee(empNr) & Date adalah unique”
- $\leq 3$  : frequency constraint,  
Artinya:  
“Tiap Parking, dibolehkan paling banyak 3 cars”

# See You Next Session

- **Thank's**