



UNIVERSITAS  
INDONESIA

*Veritas, Probitas, Justitia*

FAKULTAS

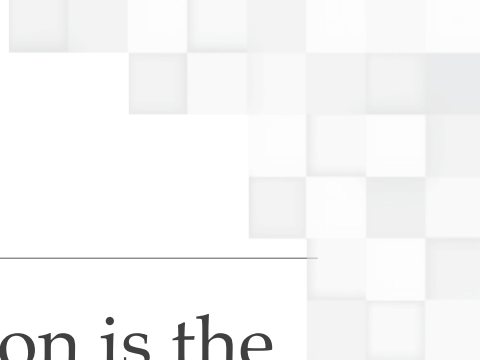
ILMU  
KOMPUTER

Slide 3

# Pemodelan Basis Data dengan ER

---

CSF2600700 - BASIS DATA  
SEMESTER GENAP 2017/2018



---

The main reference of this presentation is the textbook and PPT from : Elmasri & Navathe, Fundamental of Database Systems, 7<sup>th</sup> edition, 2015, Chapter 7

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

# Tujuan Pembelajaran

---

Mempelajari model konseptual dari basis data dengan *Entity Relationship Diagram* (ERD)

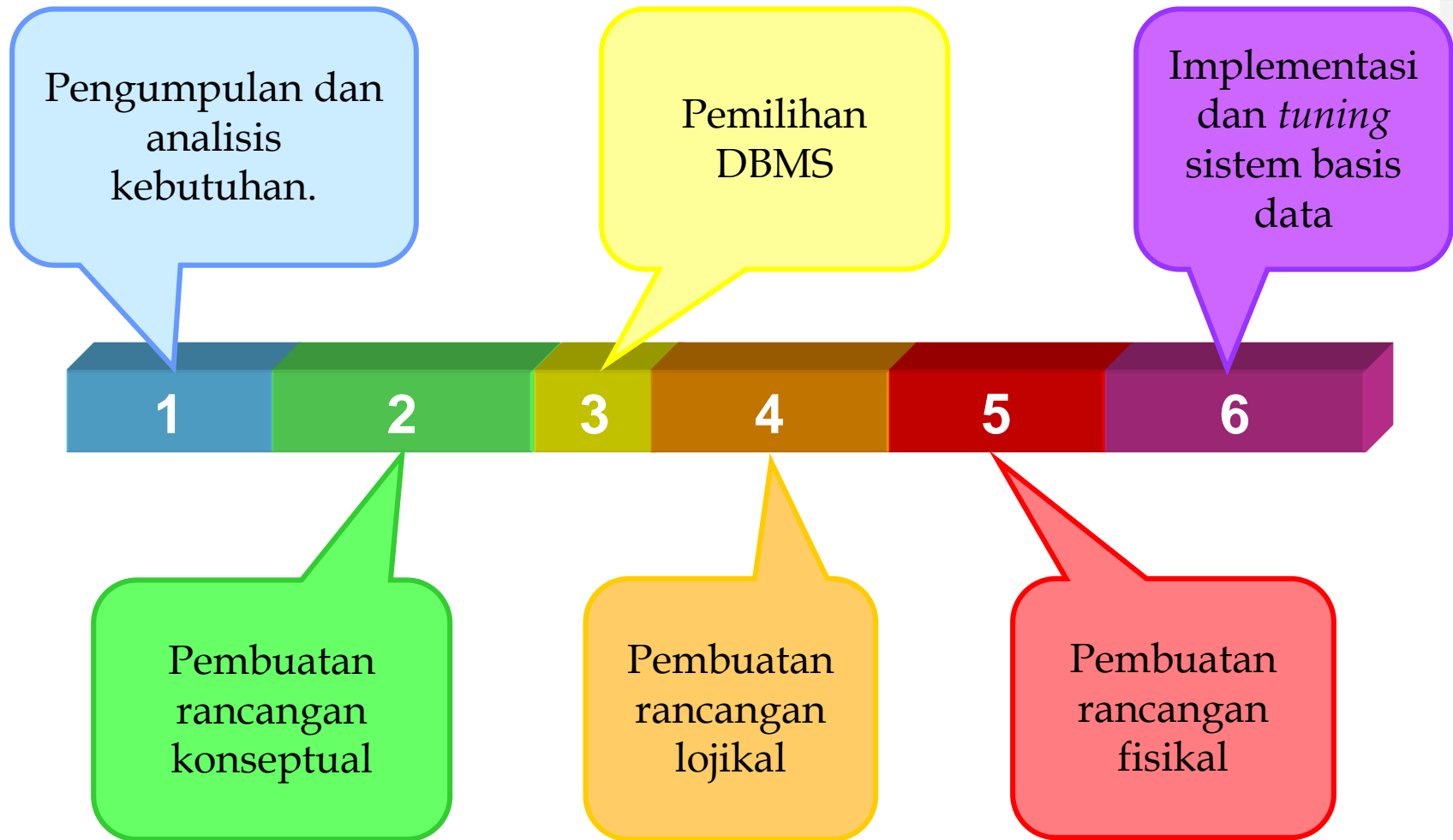
- Notasi dari ERD
- Identifikasi dan memodelkan entitas
- Identifikasi dan memodelkan relasi antar entitas

# Outline

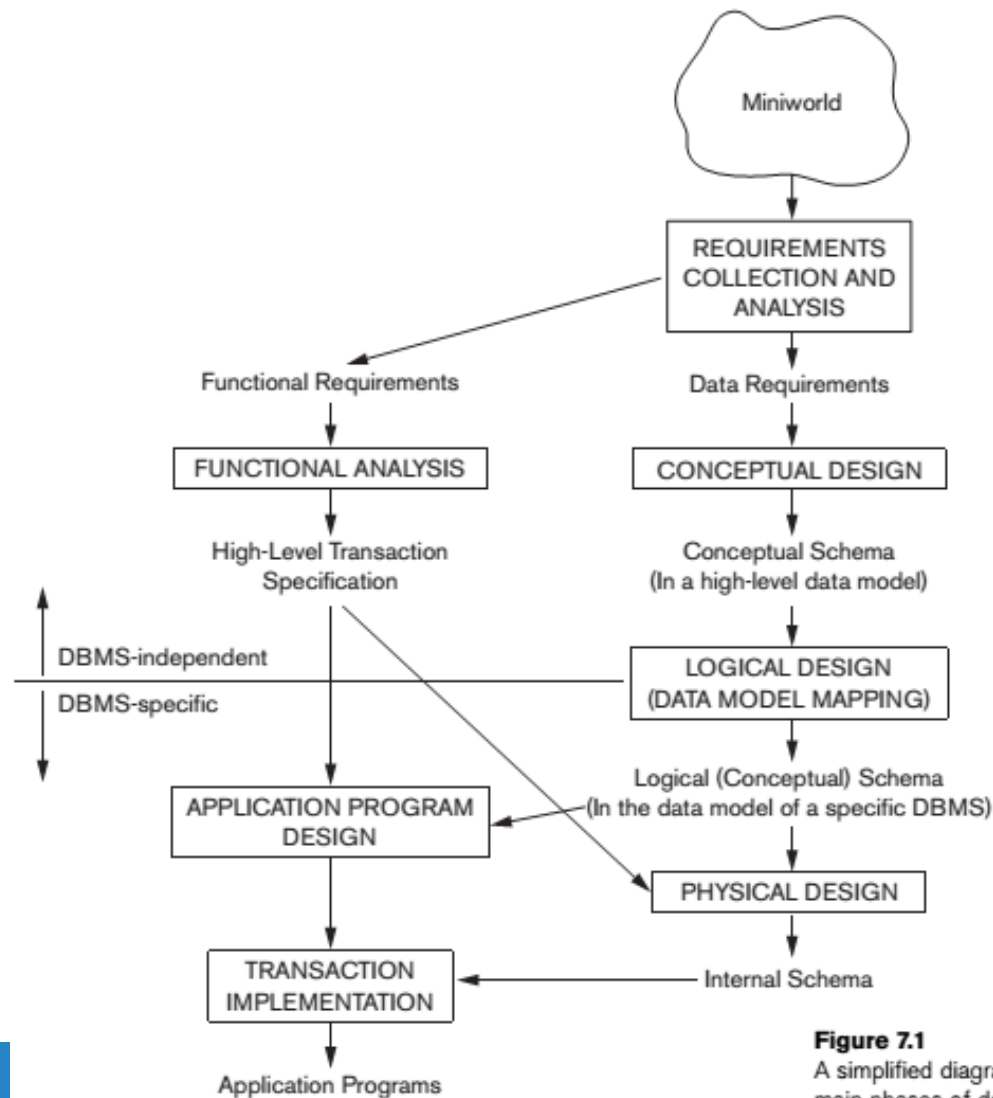
---

Tahapan Pengembangan Basis Data  
Contoh Aplikasi Basis Data  
Konsep-konsep model ER  
Pembuatan diagram ER  
Keterbatasan diagram ER

# Tahapan Pengembangan Basis Data



# Tahapan Pengembangan Basis Data



**Figure 7.1**

A simplified diagram to illustrate the main phases of database design.

# Skema Bisnis

---

Tahap pengumpulan dan analisis *requirement* menghasilkan skema bisnis organisasi/perusahaan.

Terdiri atas:

- Aktivitas bisnis, cakupan bisnis, dan semua aspek di dalamnya
- Struktur organisasi
- Kebijakan, standar serta prosedur operasional untuk mengatur aktivitas bisnis

ERD merupakan diagram skema bisnis



UNIVERSITAS  
INDONESIA

*Veritas, Profectus, Justitia*

FAKULTAS  
**ILMU  
KOMPUTER**

# Contoh Aplikasi Basis Data

---



# *Requirements* Basis Data COMPANY

---

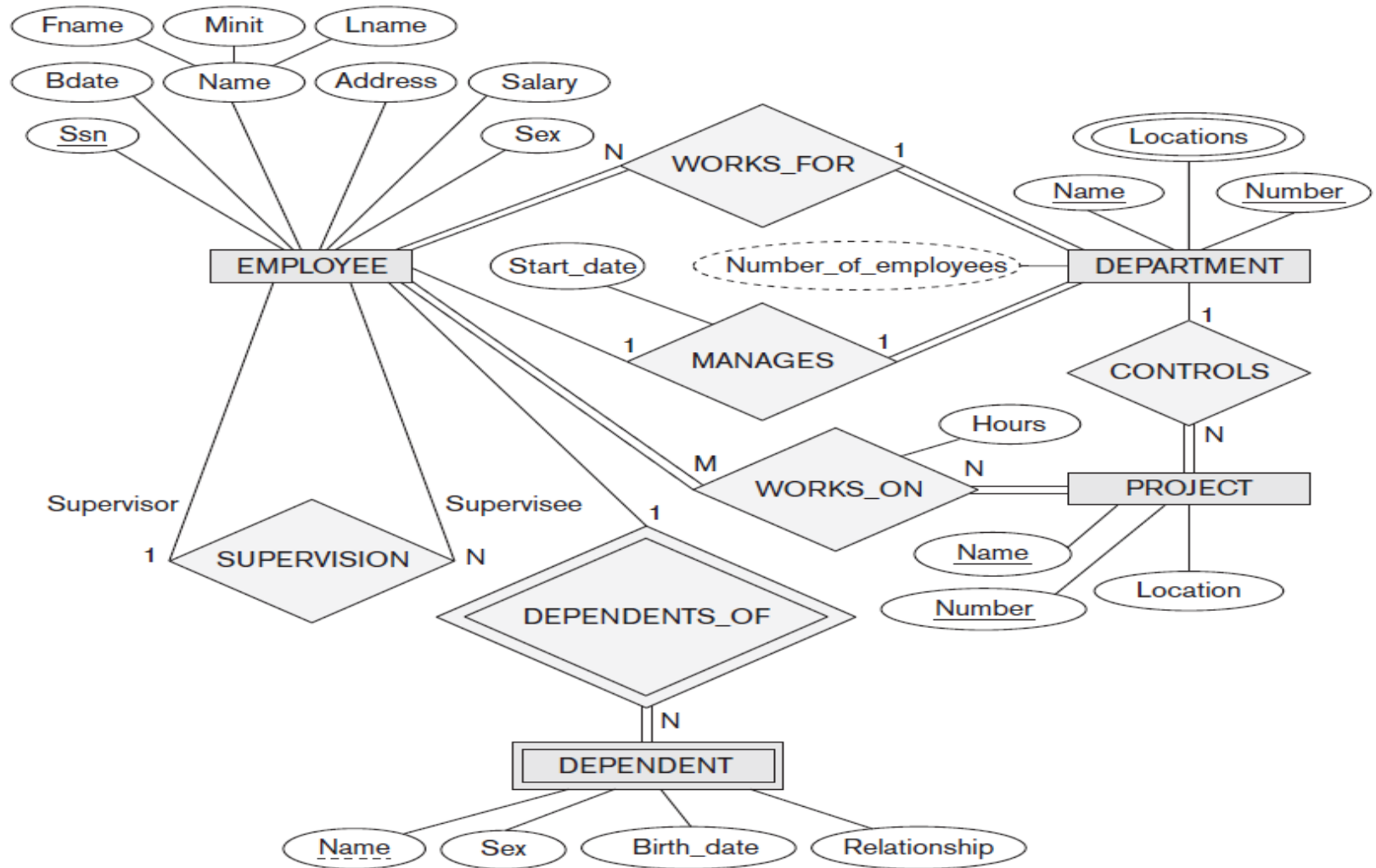
- ◆ Company memiliki beberapa DEPARTMENTS
  - Tiap DEPARTMENT memiliki nama, nomor, lokasi dan seorang pegawai yang **me-manage** DEPARTMENT
  - Satu DEPARTMENT dapat berlokasi di beberapa tempat
  - Tanggal mulai kerja dari manajer perlu disimpan
- ◆ Tiap DEPARTMENT **mengontrol** PROJECTs
  - Tiap PROJECT memiliki nama, nomor dan berlokasi di satu tempat

# *Requirements* Basis Data COMPANY

---

- ◆ Data pribadi PEGAWAI meliputi SSN, name (first name, middle name, last name), address, salary, sex, birthdate
  - Tiap EMPLOYEE bekerja pada satu DEPARTMENT, namun dapat terlibat pada beberapa PROJECT, yang tidak harus di department yang sama dengan pegawai.
  - Jumlah jam kerja pegawai pada tiap project dicatat
  - Supervisor langsung dari setiap EMPLOYEE juga dicatat
- ◆ Pegawai dapat memiliki tanggungan (DEPENDENTs)
  - Dicatat nama, jenis kelamin, tanggal lahir, dan hubungan tanggungan dengan pegawai

# Diagram ER untuk basis data COMPANY



**Figure 7.2**

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.



UNIVERSITAS  
INDONESIA

*Veritas, Profectus, Justitia*

FAKULTAS  
**ILMU  
KOMPUTER**

# Konsep dan Notasi Model ER

---

# Komponen Diagram ER

---

*Entity*

*Attribute*

*Relationship (antar-entity)*

- *Relationship types*
- *Role*
- *Cardinality*

# *Entity*

---

*Entity* adalah objek atau benda tertentu dalam *miniworld* yang direpresentasikan pada basis data.

*Entity* dapat berupa:

- Benda yang ada secara fisik (orang, mobil, rumah, dll)
- Benda yang ada secara konseptual (perusahaan, pekerjaan, mata kuliah, dll)

# Contoh *Entity*

Entity dapat berupa *person*, tempat, objek, *events*, konsep dimana kita *meng-capture* dan menyimpan data

## ***Person***

Agen, kontraktor, pembeli, departemen, divisi, pegawai, guru, siswa, *supplier*

## **Tempat**

Area penjualan, gedung, ruang, kantor cabang, kampus

## **Objek**

Buku, mesin, produk, material, lisensi software, paket software, alat, kendaraan

## ***Event***

penerbangan, penagihan, pemesanan, lomba, perjalanan, penjualan, pembatalan

## **Konsep**

akun, blok waktu, kuliah, kualifikasi, stok

# *Attribute*

*Attribute* adalah sifat-sifat yang digunakan untuk menerangkan *entity*

Suatu *entity* tertentu akan memiliki nilai untuk tiap atributnya.

- Contoh: *employee entity* tertentu memiliki Name='Budi', SSN='123456789', Address ='Depok', Sex='M', BirthDate='05-JAN-55'

Setiap *attribute* memiliki himpunan nilai yang berasosiasi dengannya.

- Disebut juga tipe data
- Contoh: integer, string, subrange, enumerated type, ...



# Type-Type *Attribute* (1)

## Simple

- Tiap *entity* memiliki nilai atomik tunggal
- Contoh: SSN, Jenis Kelamin.

## Composite

- *Attribute* terdiri dari beberapa komponen
- Contoh:
  - Address (Apt#, House#, Street, City, State, ZipCode, Country)
  - Name (FirstName, MiddleName, LastName).
- Komposisi dapat berupa hirarki dimana komponen *attribute* juga merupakan composite *attribute*.

## Multi-valued

- Sebuah *entity* dapat memiliki beberapa nilai
- Contoh:
  - Warna dari sebuah MOBIL
  - Gelar dari PEGAWAI
  - Dinyatakan dalam: {Warna}, {Gelar}.

## Time-Type Attribute (2)

*Attributes composite* dan *multi-valued* dapat dibuat bersarang (*nested*)

- {PreviousDegrees (College, Year, Degree, Field)}.

### *Stored vs. Derived Attribute*

- *Stored*: regular attribute
- *Derived*: attribute yang diturunkan/ dikalkulasi dari *stored attribute*
- Contoh: BirthDate vs Age

# Type-Type *Attribute* (3)

---

## Nilai NULL

- “*nothing*”, bukan nol, bukan spasi kosong!

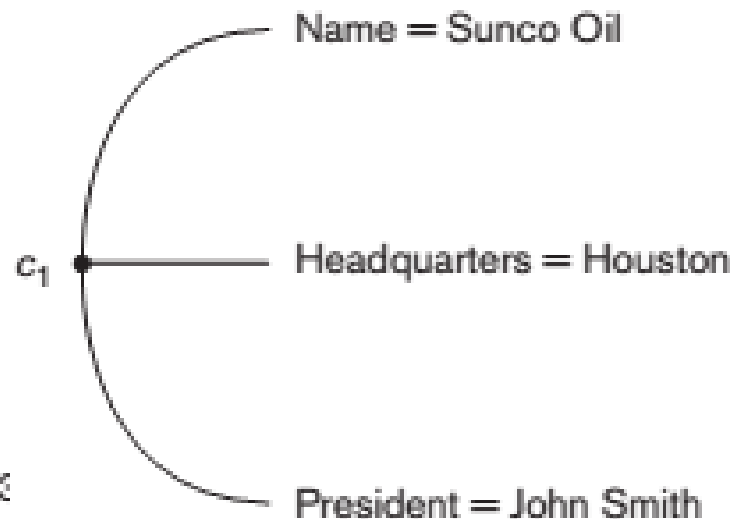
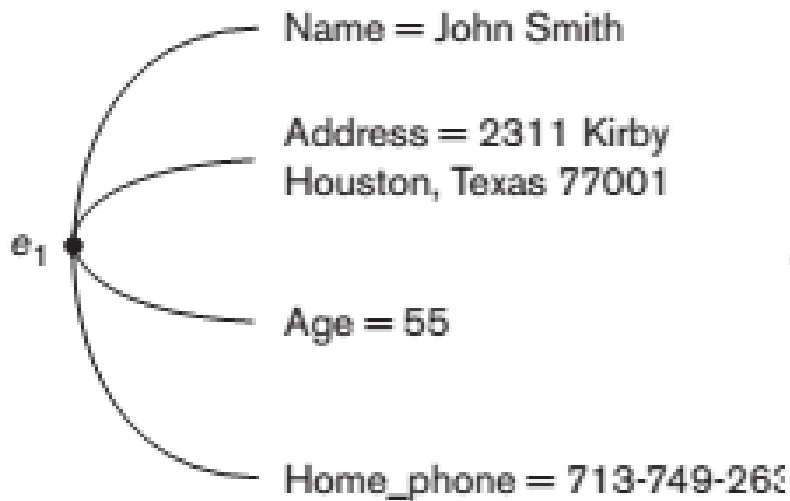
## Mengapa *attribute* dapat bernilai *null*?

- *Not applicable*, misal nama pasangan untuk orang yang belum menikah
- *Unknown*:
  - *Missing*: nilainya ada tapi belum diketahui, misalnya: tinggi badan seseorang.
  - *Not known*: Anda tidak tahu apakah nilai tersebut ada. Misalnya: nomor telepon rumah.

# Contoh *Entity* beserta Nilai *Attribute*-nya

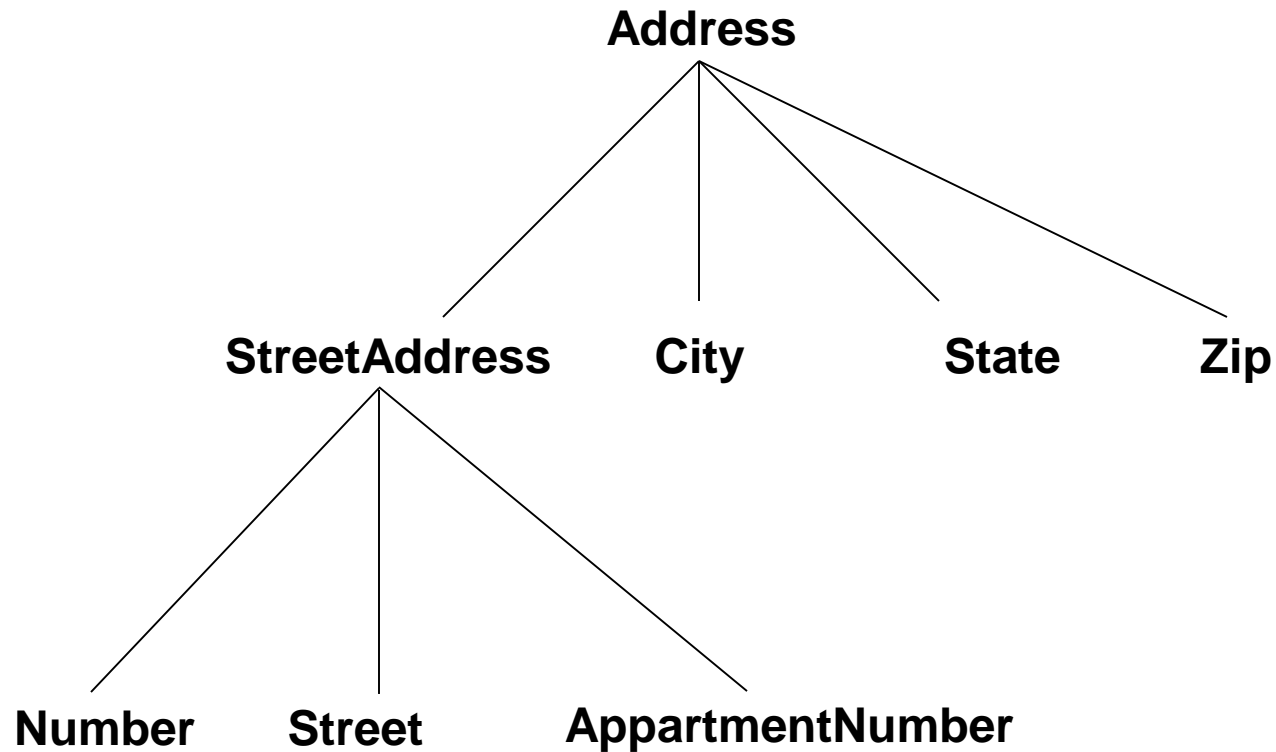
$e_1$ : employee

$c_1$ : company



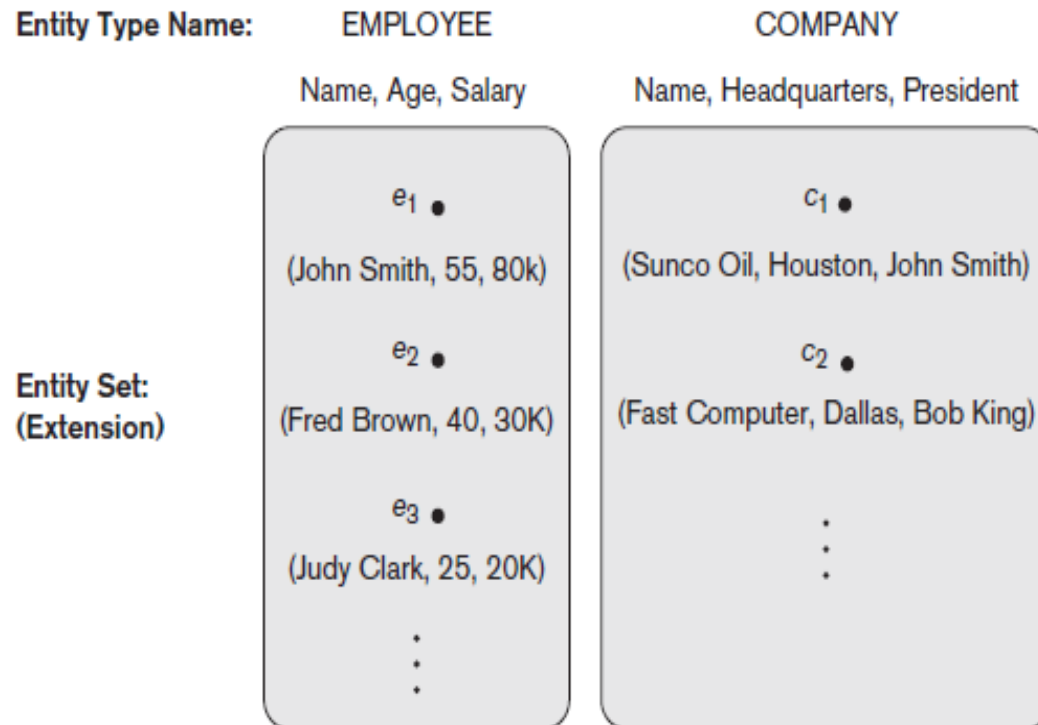
**Figure 7.3**  
Two entities, EMPLOYEE  $e_1$ , and COMPANY  $c_1$ , and their attributes.

# Contoh Hirarki dari *Composite Attributes*



# Entity Type

Mendefinisikan suatu himpunan *entity* dengan kesamaan *attributes*



**Figure 7.6**

Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

# *Key Attribute dari Entity Type*

---

Apa yang membuat suatu *entity* menjadi unik?

- An employee: SSN
- A company: name
- A project: number, name
- A purchase slip: date & time

# Pemilihan *Key Attribute*

---

Pemilihan *key attributes* merupakan hal yang penting dalam perancangan basis data

- Mempengaruhi validitas integritas data dan kinerja basis data
- Dengan mendeklarasikan suatu *attribute* sebagai *key* dan mendeklarasikan '*duplicates not allowed*' akan mencegah *user* memasukkan data yang sama yang tidak diinginkan
- Sebuah *key* juga akan memelihara integritas dengan cara menghubungkan *key* ini dengan *key* pada tabel lain



# Aturan dalam Pemilihan *Key Attribute*

---

*Key* merupakan *attribute* yang tidak berubah

- Contoh: SSN, employee\_ID, license plate number

*Key* tidak dapat bernilai *null*, harus punya nilai yang valid

- Contoh: Tanggal kelulusan siswa merupakan pilihan yang buruk untuk dijadikan *key*

Hindari *key* yang memiliki *intelligence* tertentu atau *codes built in*

- Contoh: Kode bangunan (yang dapat berubah di kemudian hari)

# *Key Attributes dan Jumlah Key*

---

*Key* dapat berupa *composite attribute*

- Contoh: VehicleTagNumber (Number, State) merupakan *key* untuk *entity type* CAR

Sebuah *entity type* dapat memiliki lebih dari satu *key*. Contoh *key* untuk CAR

- VehicleIdentificationNumber (yang sering disebut *VIN*) dan
- VehicleTagNumber (Number, State), juga dikenal sebagai *license plate number*

# *Requirements* Basis Data COMPANY

---

- ◆ Company memiliki beberapa DEPARTMENTS
  - Tiap DEPARTMENT memiliki nama, nomor, lokasi dan seorang pegawai yang **me-manage** DEPARTMENT
  - Satu DEPARTMENT dapat berlokasi di beberapa tempat
  - Tanggal mulai kerja dari manajer perlu disimpan
- ◆ Tiap DEPARTMENT **mengontrol** PROJECTs
  - Tiap PROJECT memiliki nama, nomor dan berlokasi di satu tempat

# *Requirements* Basis Data COMPANY

---

- ◆ Data pribadi PEGAWAI meliputi SSN, name (first name, middle name, last name), address, salary, sex, birthdate
  - Tiap EMPLOYEE bekerja pada satu DEPARTMENT, namun dapat terlibat pada beberapa PROJECT
  - Jumlah jam kerja pegawai pada tiap project dicatat
  - Supervisor langsung dari setiap EMPLOYEE juga dicatat
- ◆ Pegawai dapat memiliki tanggungan (DEPENDENTs)
  - Dicatat nama, jenis kelamin, tanggal lahir, dan hubungan tanggungan dengan pegawai

# Awal Konsep *Entity* untuk Basis Data COMPANY

---

## DEPARTMENT

Name, Number, Locations, Manager, ManagerStartDate

## PROJECT

Name, Number, Location, ControllingDepartment

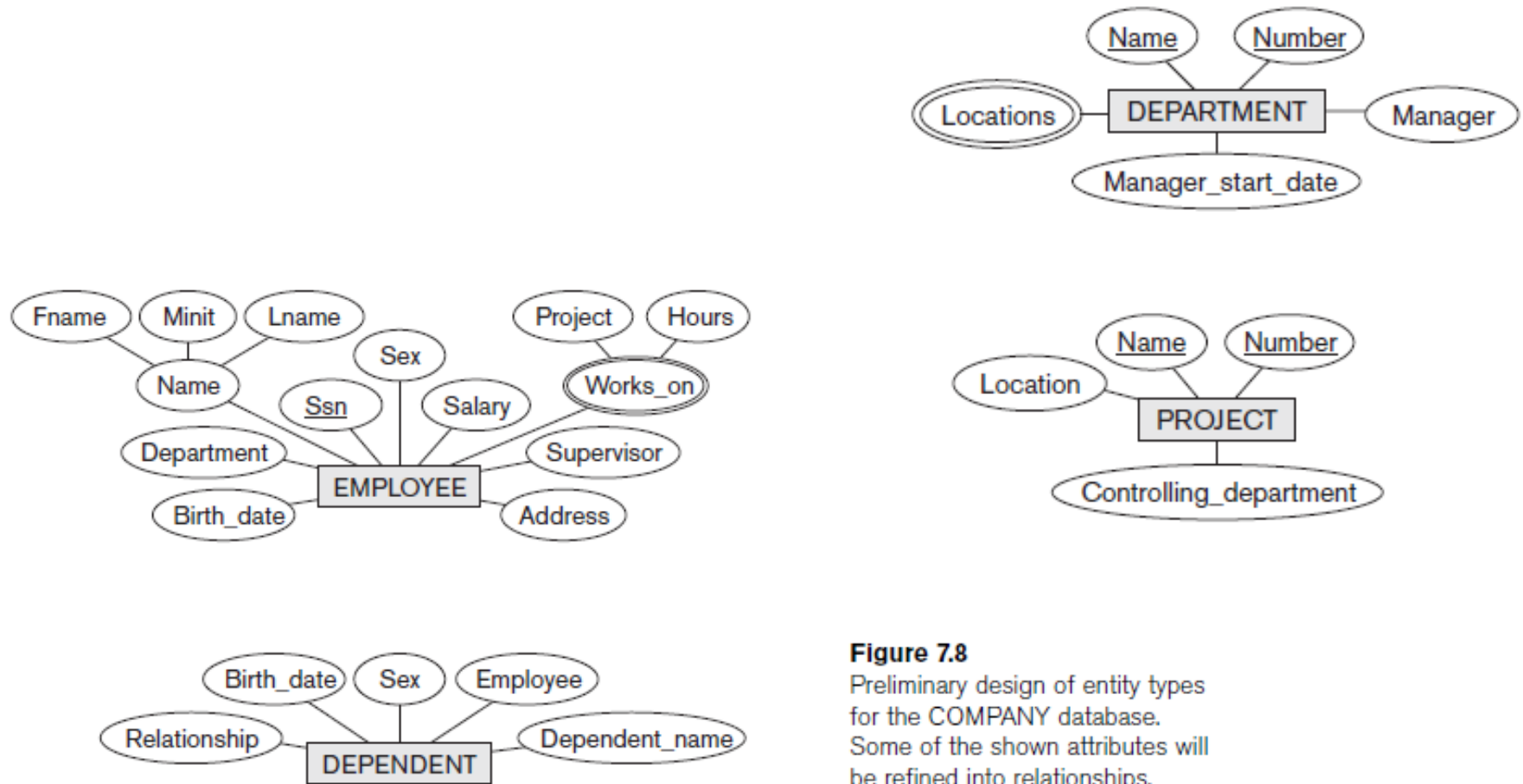
## EMPLOYEE

Name (ENAME, MInit, LName), SSN, Sex, Address,  
Salary, BirthDate, Department, Supervisor, {Workson  
(Project, Hours)}

## DEPENDENT

Employee, DependentName, Sex, Birthdate,  
Relationship

# Awal Konsep *Entity* untuk Basis Data COMPANY



**Figure 7.8**

Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

# Relationship

Sebuah *relationship* menghubungkan dua/lebih *entity* yang berbeda dengan makna tertentu. Contoh:

- EMPLOYEE John Smith *works on* the ProductX PROJECT
- EMPLOYEE Franklin Wong *manages* the Research DEPARTMENT.

*Relationships* yang bertipe sama dikelompokkan ke dalam sebuah *relationship type*. Contoh:

- **WORKS\_ON** *relationship type in which EMPLOYEES and PROJECTs participate*
- **MANAGES** *relationship type in which EMPLOYEES and DEPARTMENTs participate.*

# *Relationships dan Relationship Types*

Derajat dari *Relationship Type* adalah jumlah *Entity Types* yang berpartisipasi pada *relationship* ini (**Binary**, **Ternary**, **N-nary**)

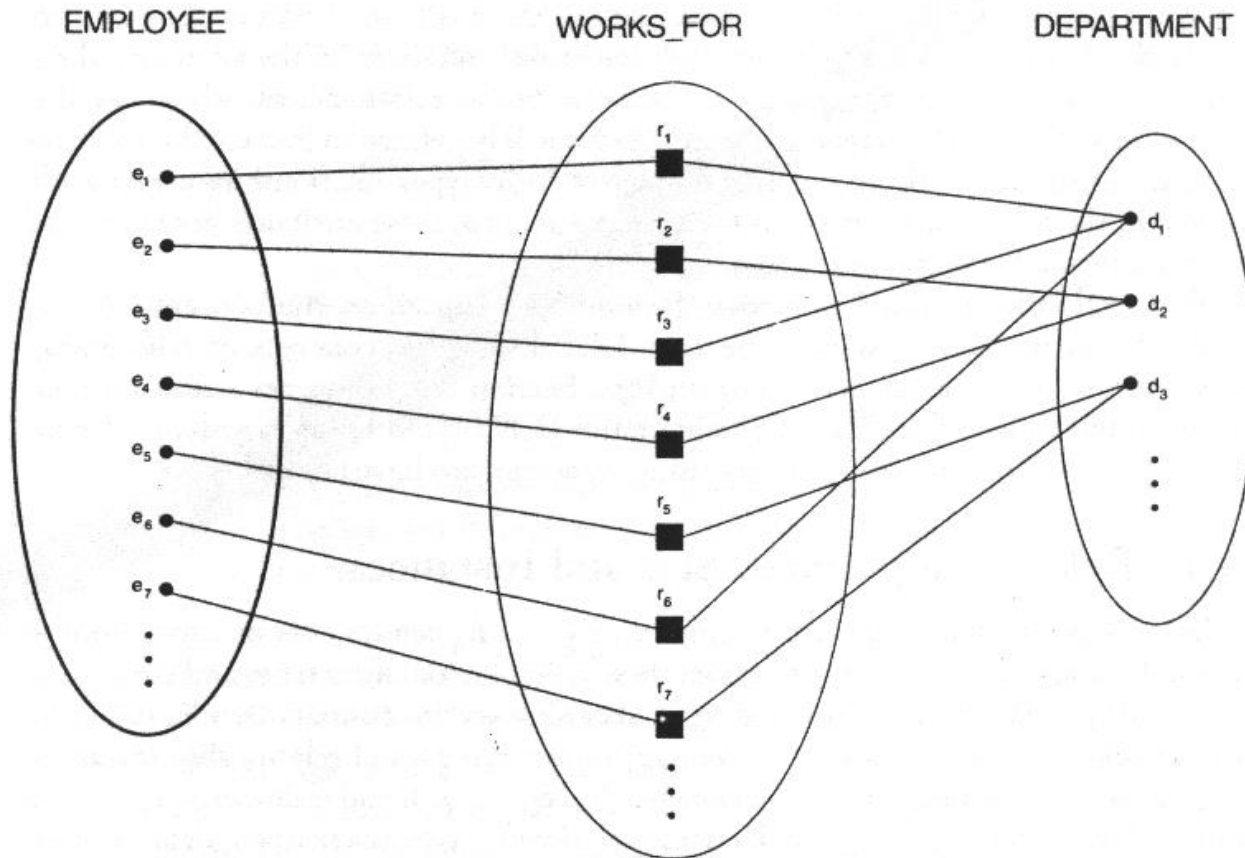
Satu *entity type* dapat berpartisipasi pada lebih dari satu *relationship type*

*Relationships* memiliki hubungan *directional* (arah tertentu):

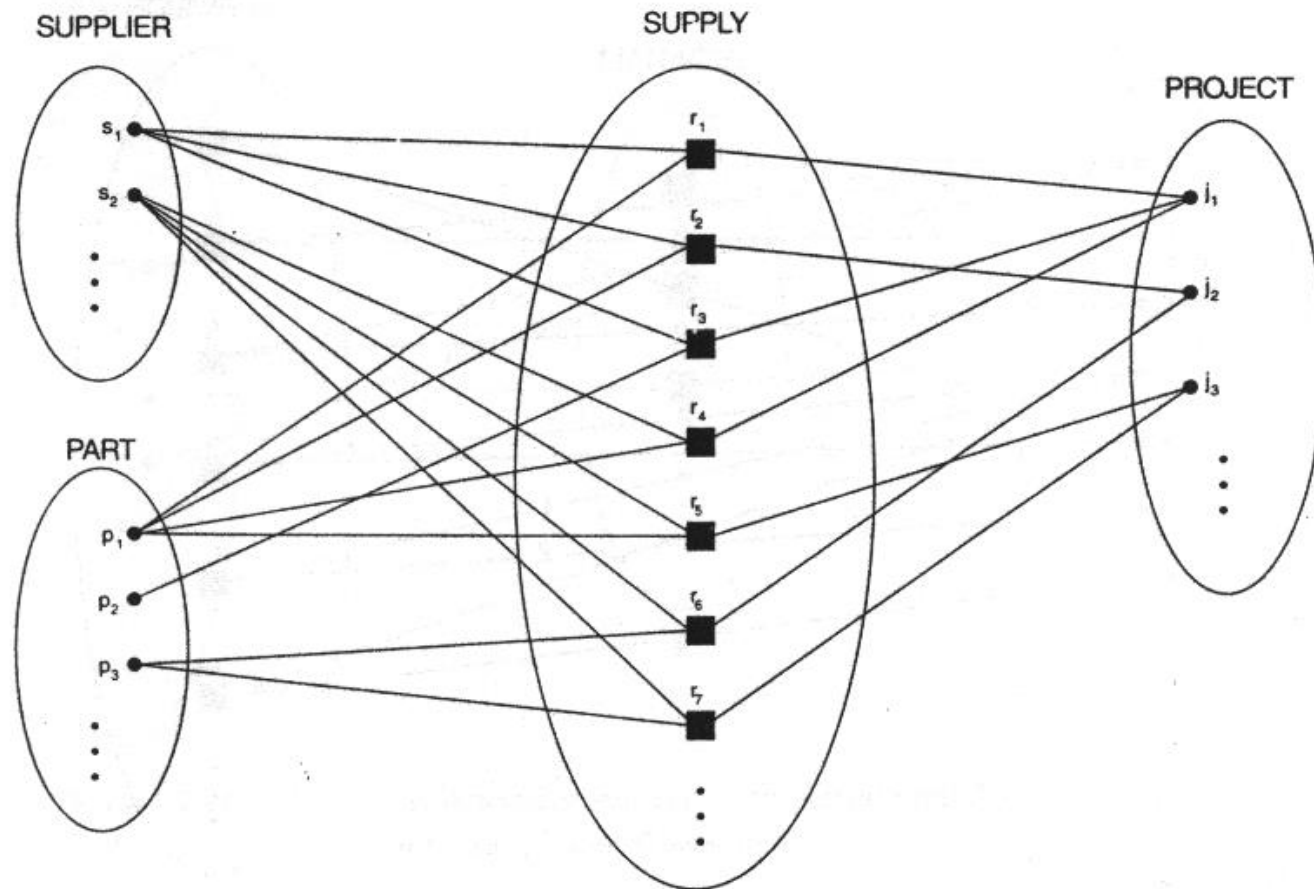
- SUPPLIES: SUPPLIER to PARTS
- SUPPLIERS: PARTS to SUPPLIER



# Binary Relationship



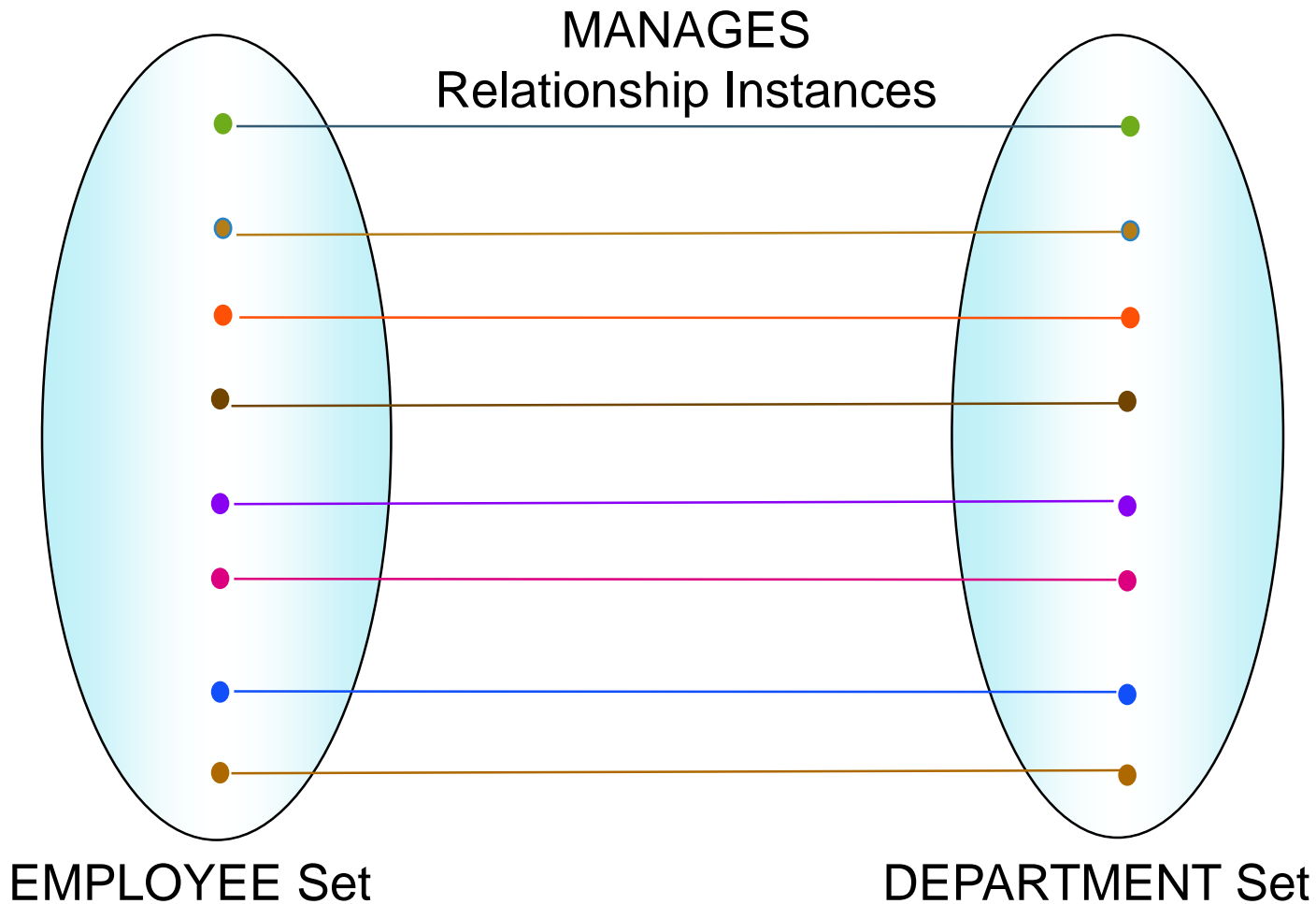
# *Ternary Relationship*



# Structural Constraints

- ◆ Digunakan untuk mengekspresikan semantik dari *relationship*. Ada 2 macam:
  - ◆ **Cardinality ratio** (dari *binary relationship*):  
1:1, 1:N, N:1, M:N
    - Ditunjukkan dengan memberikan nomor yang sesuai pada *link* dari *relationship*
  - ◆ **Participation constraint**: *total, partial*
    - *Double link* untuk *total participation*
    - *Single link* untuk *partial participation*

# *One-to-One (1:1) Relationship*

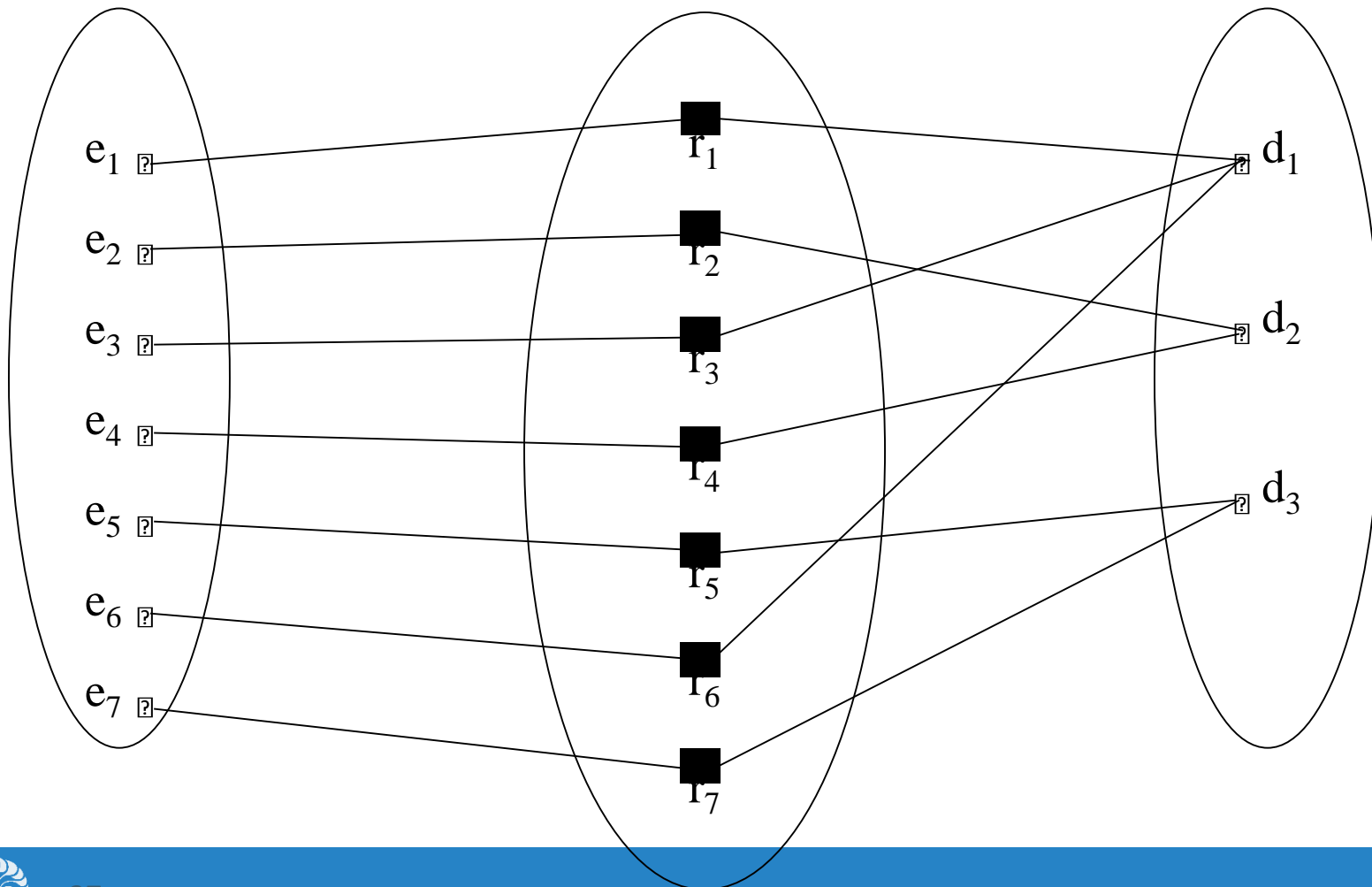


# Many-to-One (N:1) Relationship

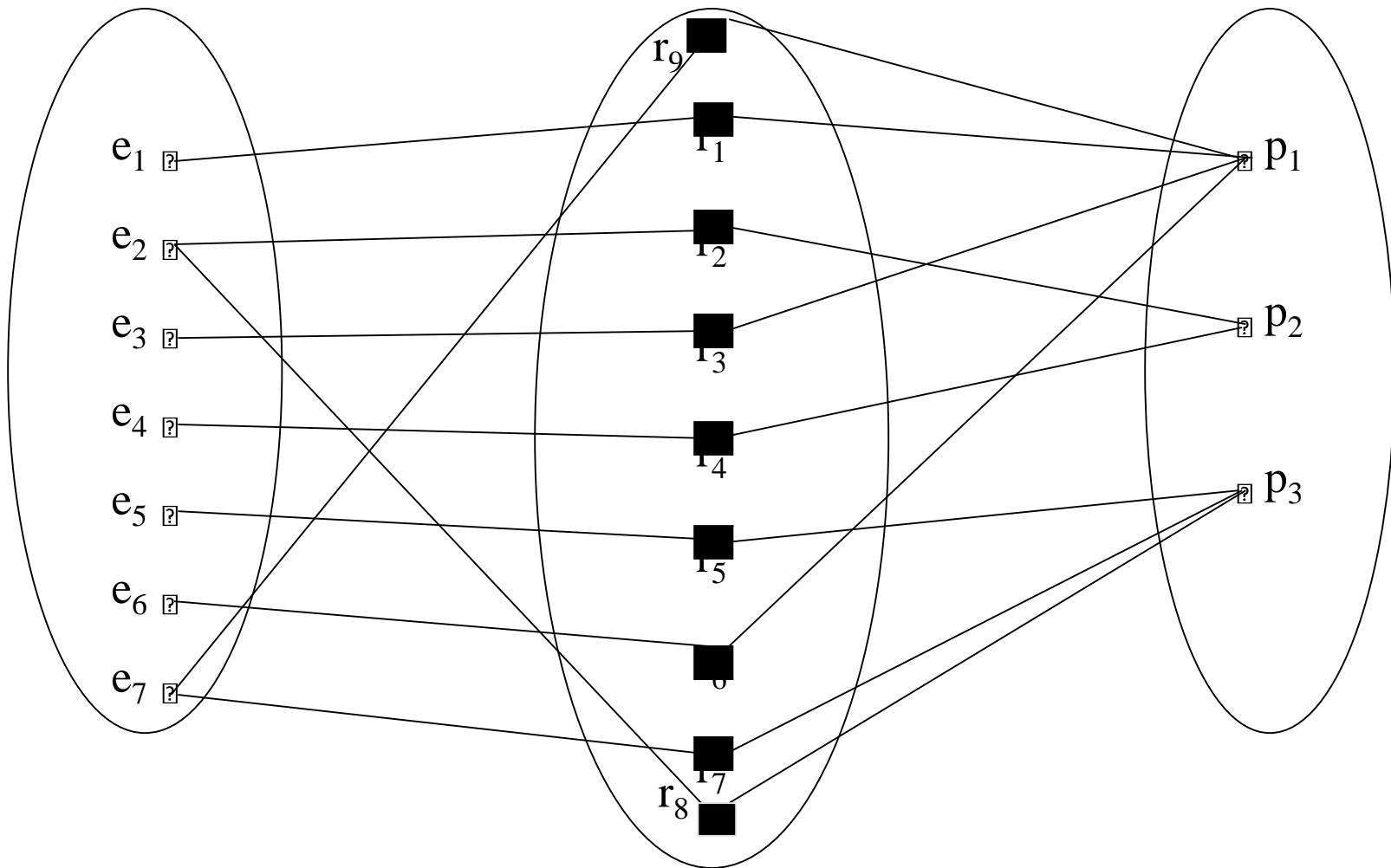
EMPLOYEE

WORKS\_FOR

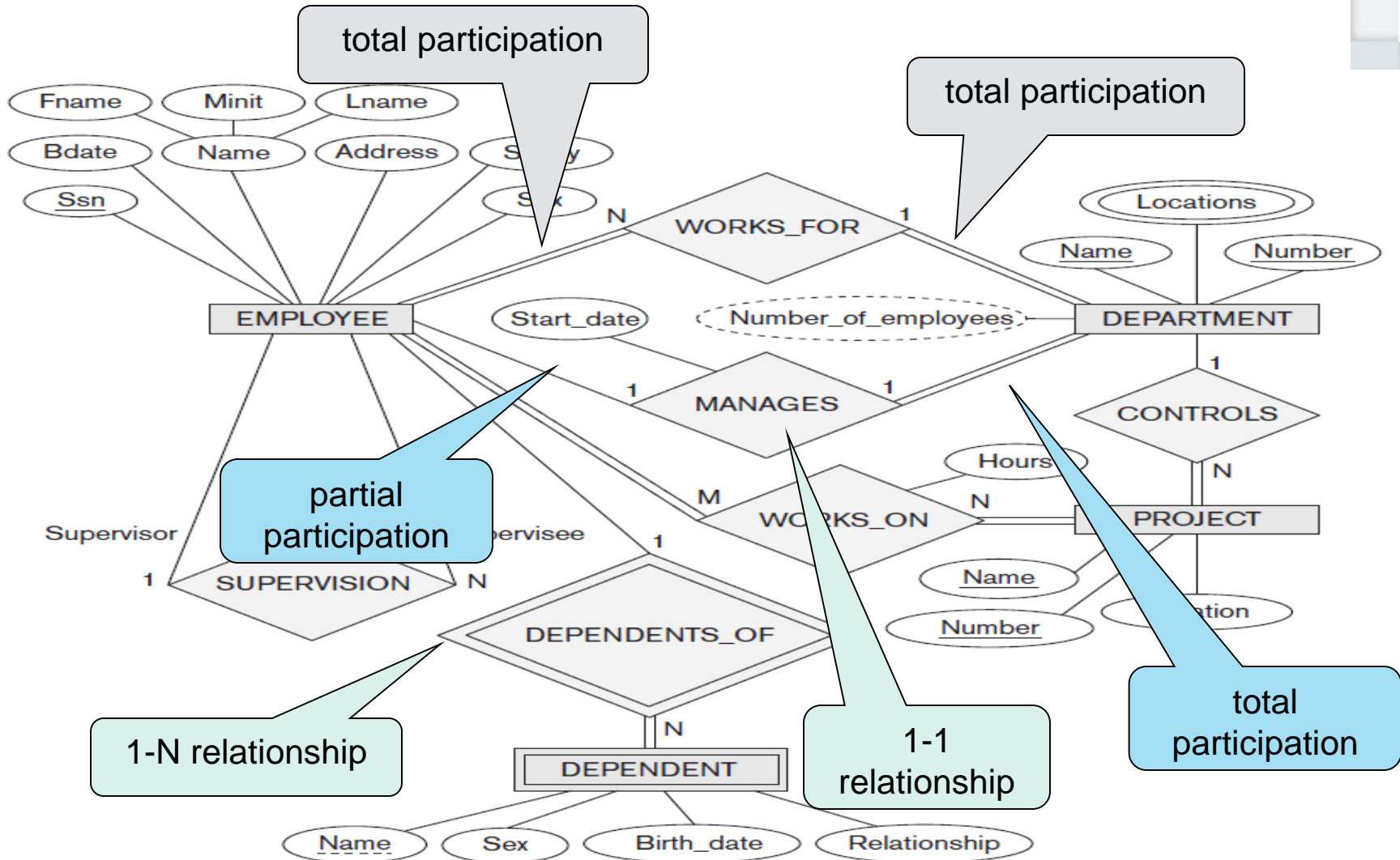
DEPARTMENT



# Many-to-Many (M:N) Relationship



# Structural Constraints



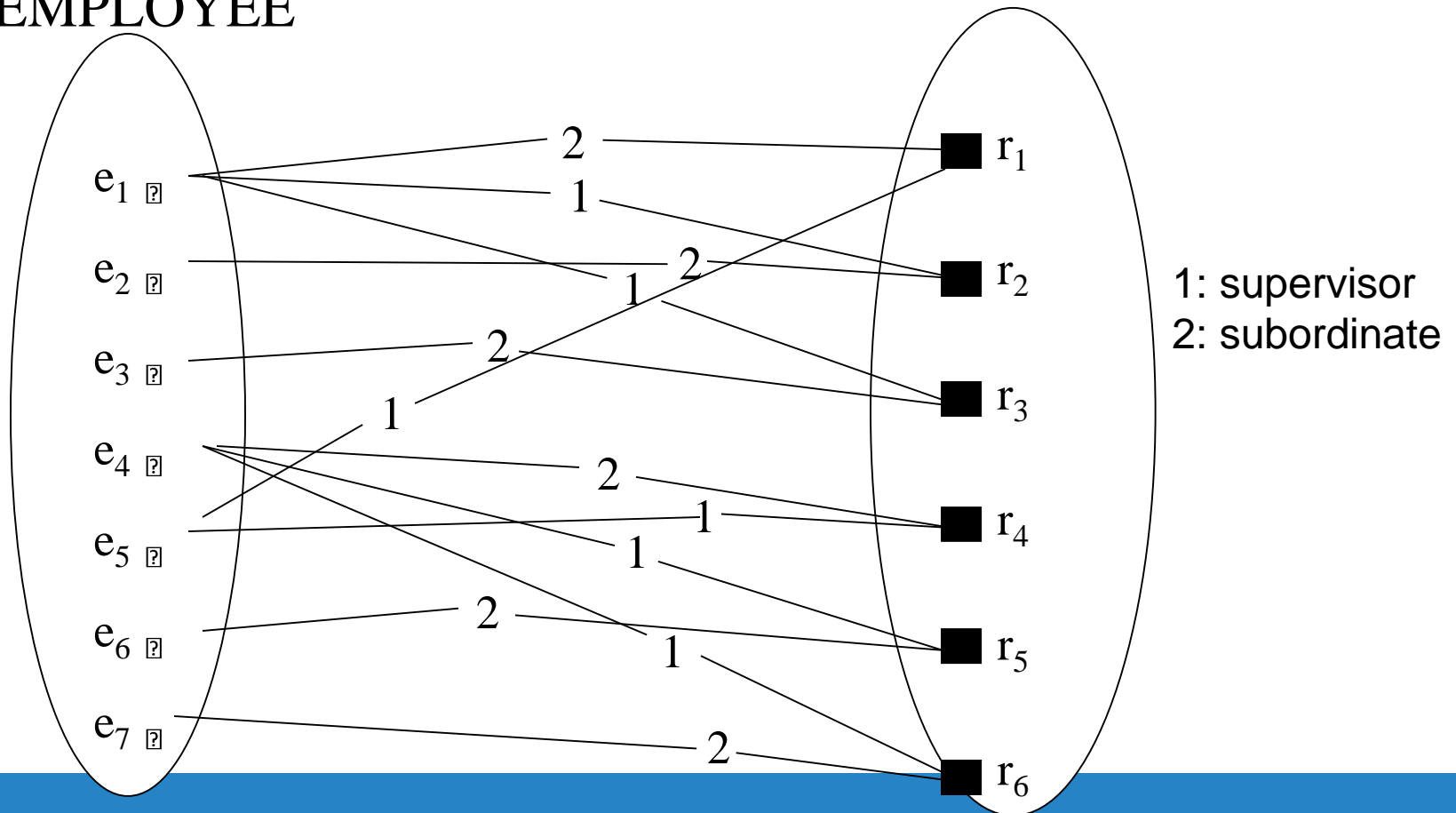
**Figure 7.2**

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

# *Recursive Relationship*

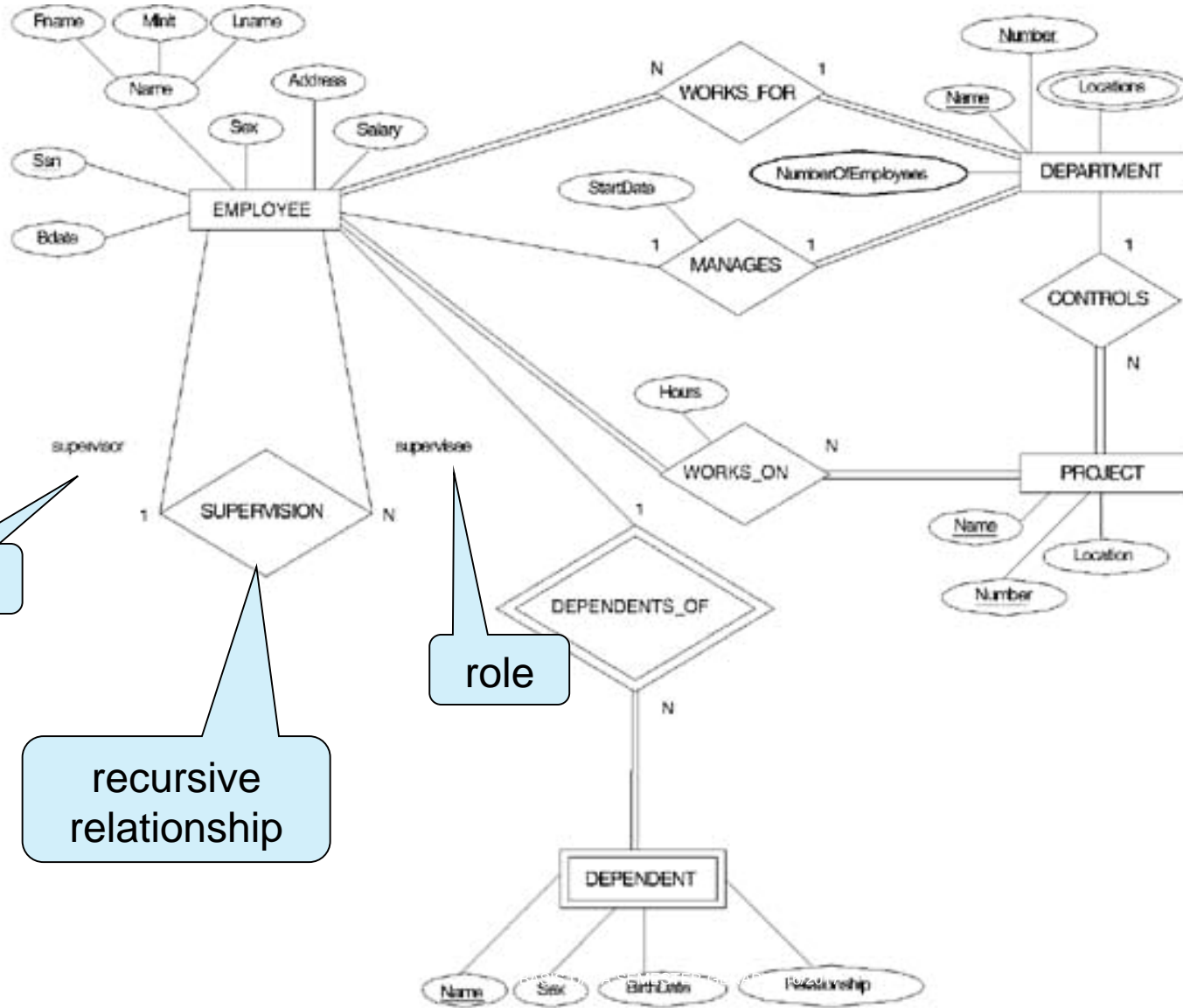
Satu *entity* berpartisipasi lebih dari 1 kali dalam *recursive relationship*, dengan *role* yang berbeda

## EMPLOYEE

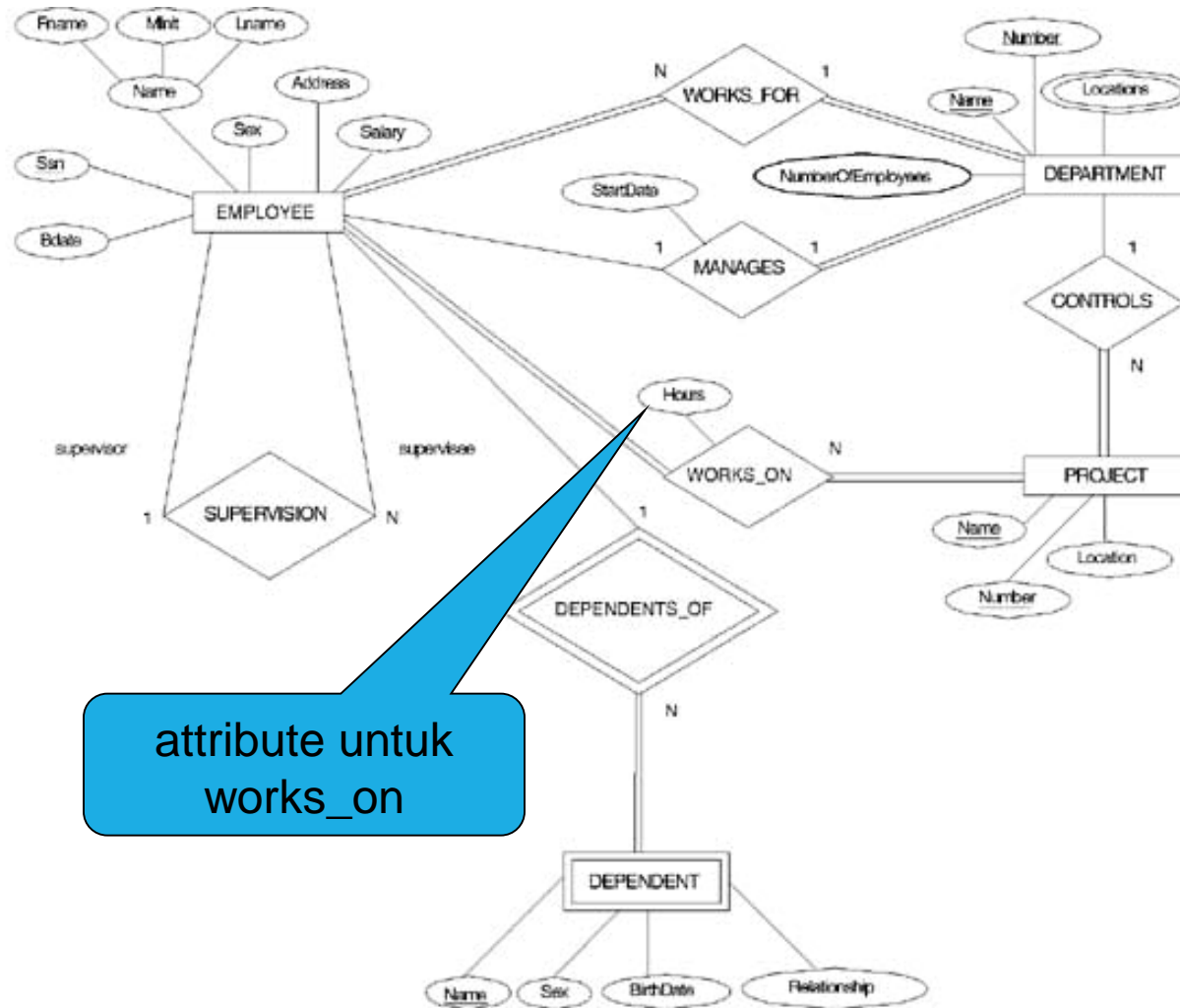




# Contoh Recursive Relationship



# Attribute dari Relationship Type



# Menentukan *Relationship*

---

Uji kombinasi dari 2 *entity*, lihat apakah ada kemungkinan *relationship* diantara keduanya.  
Lihat dokumen *requirements*

# *Weak Entity Types*

---

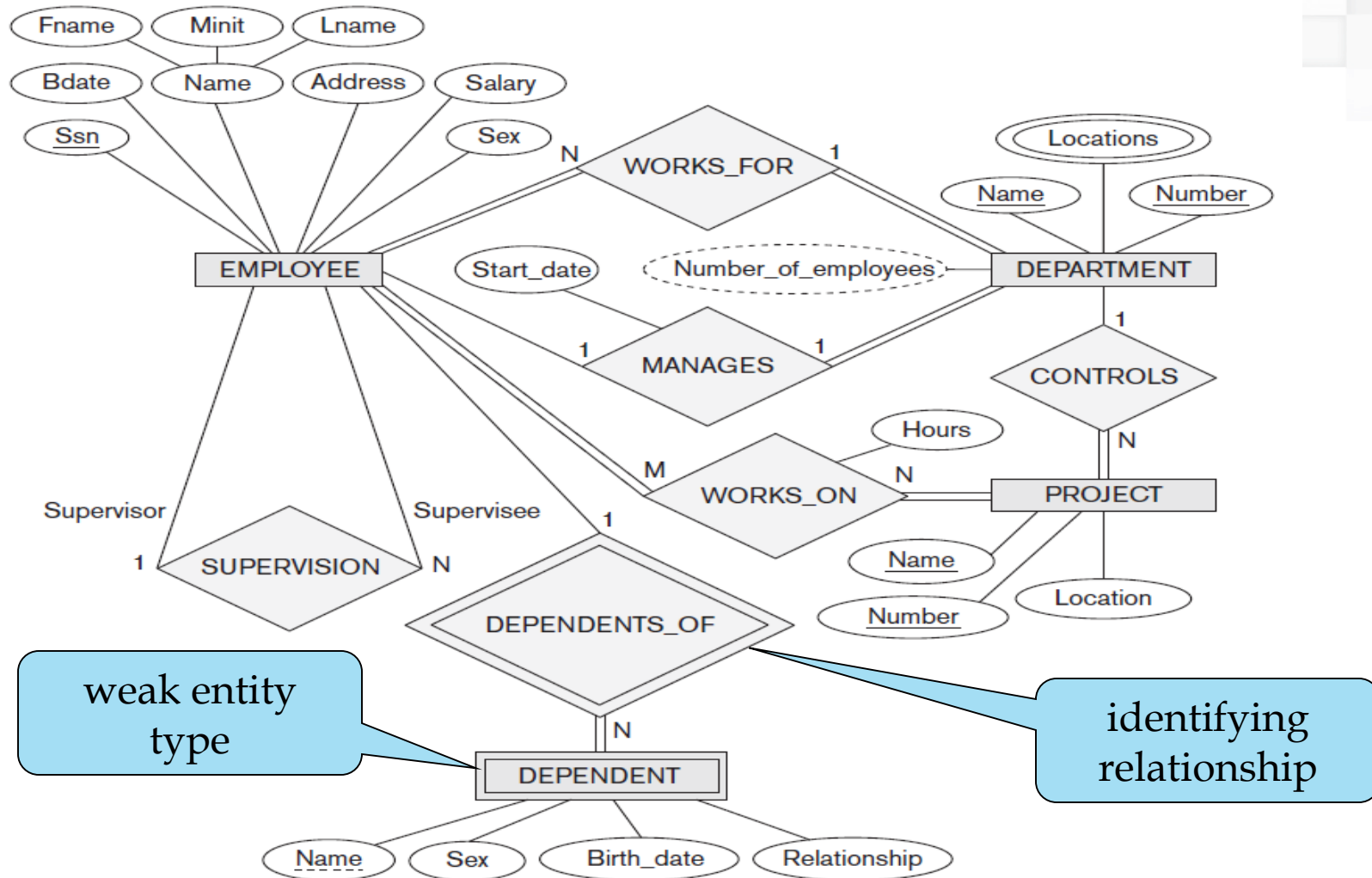
Merupakan *entity* yang tidak punya *key attribute*

*Weak entity* harus berpartisipasi dalam sebuah *identifying relationship type* dengan suatu *owner* atau *identifying entity type*

*Weak entity* diidentifikasi dengan kombinasi dari:

- *Partial key* dari *weak entity type*
- *Attribute* pada *owner entity* yang menjadi penghubung antara *weak entity* dengan *owner-nya*.

# Contoh Weak Entity Types



**Figure 7.2**

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

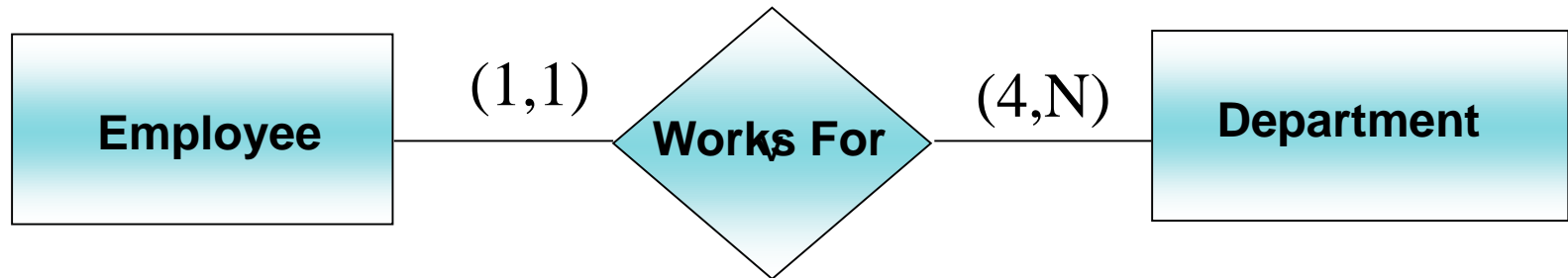
# Alternatif Notasi (min, max) untuk *Relationship Structural Constraints*

- ◆ Dinyatakan pada setiap partisipasi dari sebuah *entity type E* pada *relationship type R*
- ◆ Menyatakan bahwa tiap *entity e* di *E* berpartisipasi paling sedikit **min** dan paling banyak **max** pada *relationship instance* di *R*
- ◆ Menyatakan batasan yang ada dalam *requirement*
- ◆ Harus memenuhi
  - $\text{min} \leq \text{max}, \text{min} \geq 0, \text{max} \geq 1$
- ◆ *Default* (tidak ada batasan)
  - $\text{min} = 0$
  - $\text{max} = n$
- ◆  $\text{min} = 0$  menandakan partisipasi parsial
- ◆  $\text{min} \geq 1$  menandakan partisipasi total

# Relationship Constraints dengan Notasi (min,max)

1 employee manages (0, 1) department

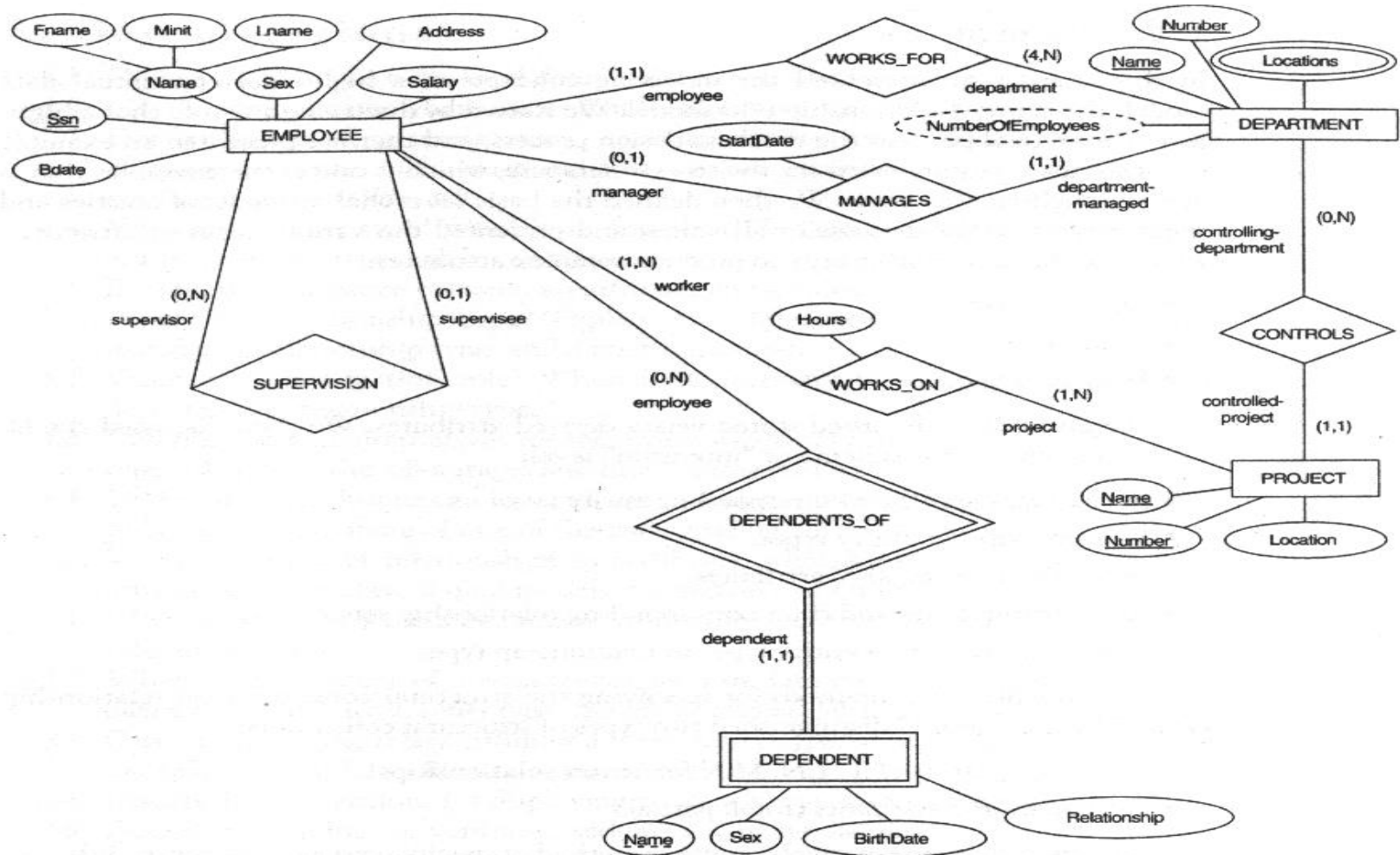
1 department is managed by (1,1) employee



1 employee works for (1, 1) department










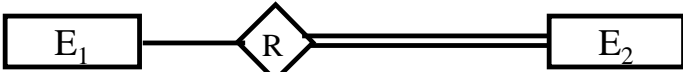


1 department employs (4,N) employees

# Diagram ER dengan Notasi min-max





# Rangkuman Notasi ER

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION OF $E_2$ IN R
	CARDINALITY RATIO 1:N FOR $E_1:E_2$ IN R
	STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R



UNIVERSITAS  
INDONESIA

*Veritas, Probitas, Justitia*

FAKULTAS

ILMU  
KOMPUTER

# Pembuatan Diagram ER

---

# Pilihan Ketika Merancang ERD

Proses pembuatan diagram ER merupakan proses yang iteratif dengan proses *refinement* yang umum.

- Adakalanya suatu konsep awalnya dimodelkan sebagai atribut, kemudian dijadikan sebagai *relationship*, karena atribut ini merupakan referensi ke *entity type* lainnya.
- Kadang-kadang, satu atribut muncul pada beberapa *entity type* dan kita pertimbangkan untuk lebih baik menjadi *entity type* tersendiri.
- Contoh: pada basis data UNIVERSITY, semula *entity* STUDENT, INSTRUCTOR & COURSE masing-masing memiliki atribut Department. Maka DEPARTMENT dapat dijadikan *entity* dengan DeptName sebagai atribut dan selanjutnya dihubungkan dengan ketiga *entity* tersebut. Atribut lain dapat dicari kemudian.

# Pilihan Ketika Merancang ERD

- Dapat pula diterapkan *refinement* yang berkebalikan dari kasus sebelumnya.
- Jika DEPARTMENT ada sebagai *entity* pada awal perancangan namun hanya memiliki 1 *attribute*, yakni STUDENT, maka DEPARTMENT dapat dipindahkan sebagai *attribute* untuk *entity* STUDENT.
- *Refinement* pada *specialization, generalization & higher degree relationship* → lihat bab selanjutnya.

# Konvensi Penamaan

---

Pilih nama (untuk *entity types*, *attributes*, *relationship types*) yang menggambarkan maknanya

Gunakan nama tunggal untuk *entity types*

Gunakan huruf kapital untuk nama *entity type* dan *relationship types*

Gunakan huruf kapital di awal kata untuk *attributes*

Gunakan huruf kecil untuk nama *role*

Nama *entity type* cenderung menggunakan kata benda

Nama *relationship types* cenderung menggunakan kata kerja

# Alat Bantu Pemodelan Data

---

Terdapat beberapa alat bantu untuk pemodelan konseptual dan pemetaan model ini ke skema relasional.

- Contoh: Ms. Visio, ER-Win, DBDesigner, S-Designer (Enterprise Application Suite), ER-Studio .
- Merupakan sarana membuat dokumentasi kebutuhan aplikasi
- Antar mukanya mudah digunakan

# Kelemahan Notasi ER

---

**Model ER tidak mendukung  
abstraksi-abstraksi yang  
berupa spesialisasi dan  
generalisasi**

**Solusi? Enhanced ER**

# Referensi

---

Elmasri & Navathe, Fundamental of Database Systems, 5<sup>th</sup> Edition, Chapter 3, 2007

Elmasri & Navathe, Fundamental of Database Systems, 6<sup>th</sup> Edition, Chapter 7, 2011



# Latihan – Soal 1

---

Mahasiswa meminjam buku.  
Tentukan *entity* dari deskripsi ini.

# Jawab: Latihan 1

---

*Entity:* MAHASISWA dan BUKU

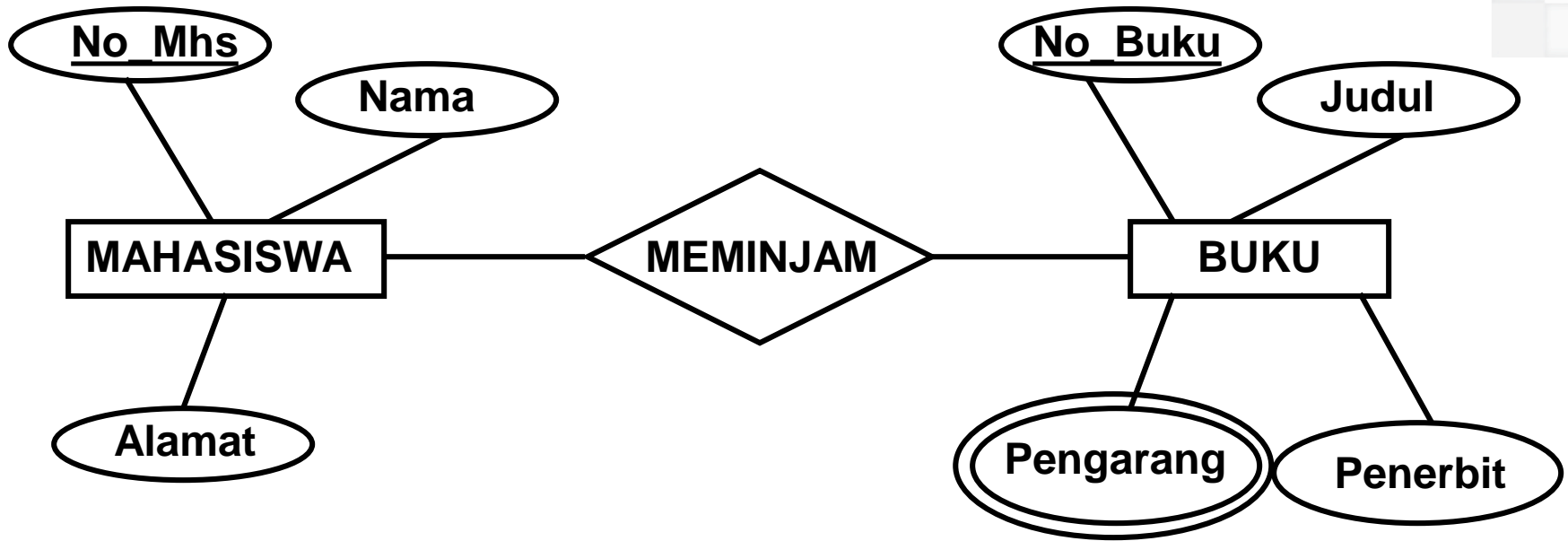
## Latihan – Soal 2

---

Pada saat mendaftar menjadi anggota perpustakaan Fakultas, dicatatlah nama, nomor mahasiswa dan alamat mahasiswa. Setelah itu mereka baru bisa meminjam buku di perpustakaan. Buku-buku yang dimiliki perpustakaan banyak sekali jumlahnya. Tiap buku memiliki data nomor buku, judul, pengarang, penerbit, tahun terbit. Satu buku bisa ditulis oleh beberapa pengarang.

Tentukan *entity*, *attribute* dan *relationship* dari deskripsi di atas.

# Jawab: Latihan 2

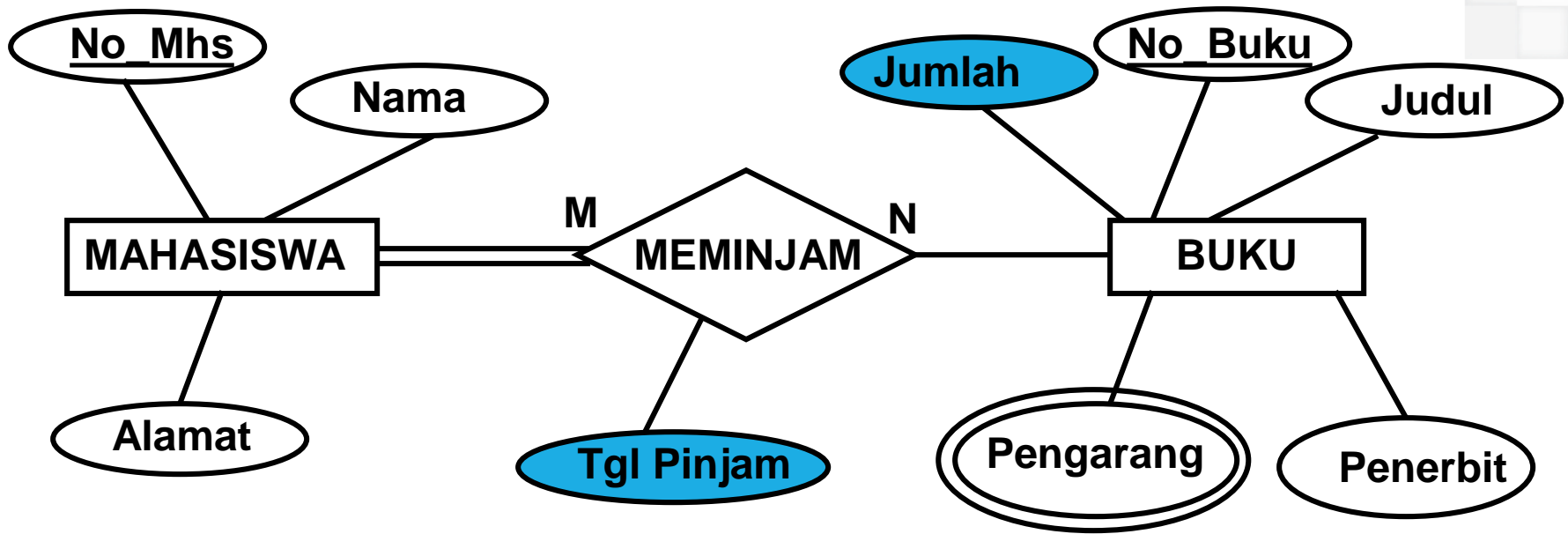


## Latihan – Soal 3

---

Seperti deskripsi soal nomor 2, namun ada beberapa tambahan penjelasan berikut ini. Seorang mahasiswa boleh meminjam beberapa buku. Satu buku boleh dipinjam beberapa mahasiswa. Semua mahasiswa sangat perlu buku sehingga tidak ada yang tidak pernah meminjam ke perpustakaan. Ada buku yang sangat laris dipinjam mahasiswa, namun ada pula buku yang tidak pernah dipinjam sama sekali. Satu buku dapat memiliki beberapa *copy*, namun untuk *copy* yang sama memiliki satu nomor buku. Setiap peminjaman akan dicatat tanggal peminjamannya. Semua mahasiswa disiplin mengembalikan buku tepat satu minggu setelah peminjaman.

# Jawab: Latihan 3

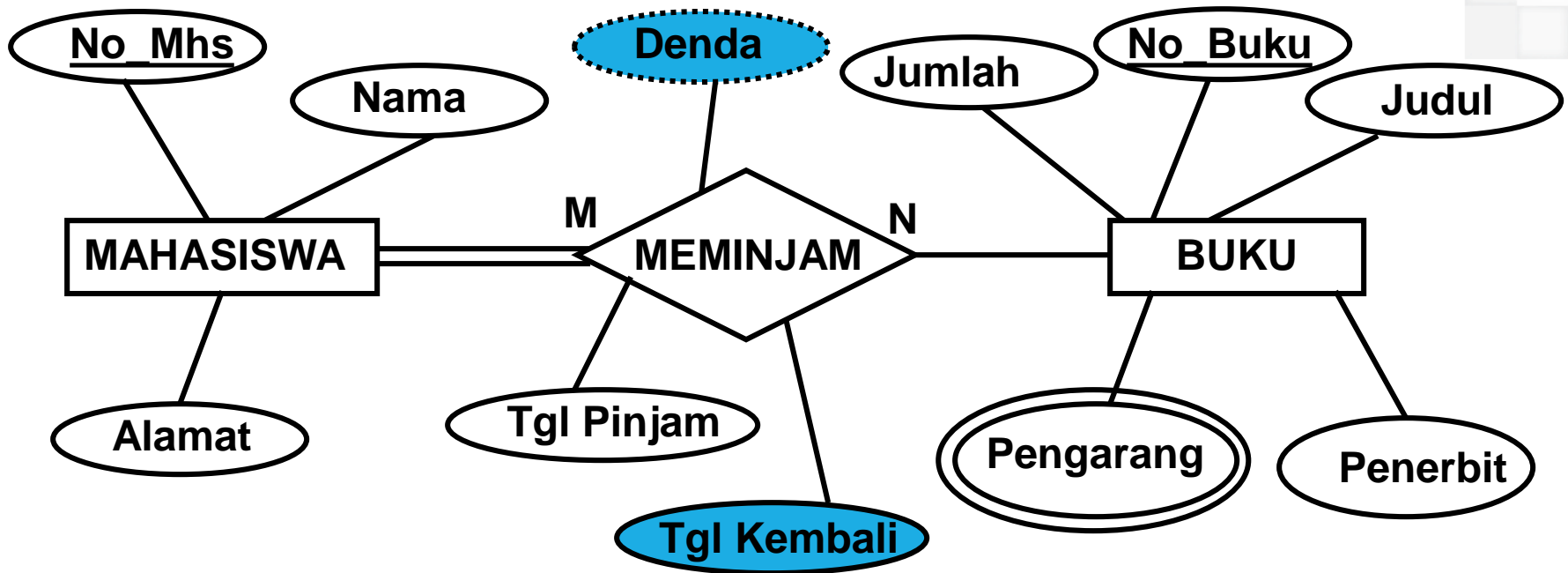


## Latihan – Soal 4

---

Seperti soal nomor 3, namun ada beberapa tambahan penjelasan berikut ini. Mahasiswa kadang-kadang terlambat mengembalikan buku, sehingga dikenakan denda. Besarnya denda adalah Rp 500,- per hari keterlambatan. Mahasiswa dianggap terlambat jika mengembalikan buku lebih lama dari 1 minggu.

# Jawab: Latihan 4



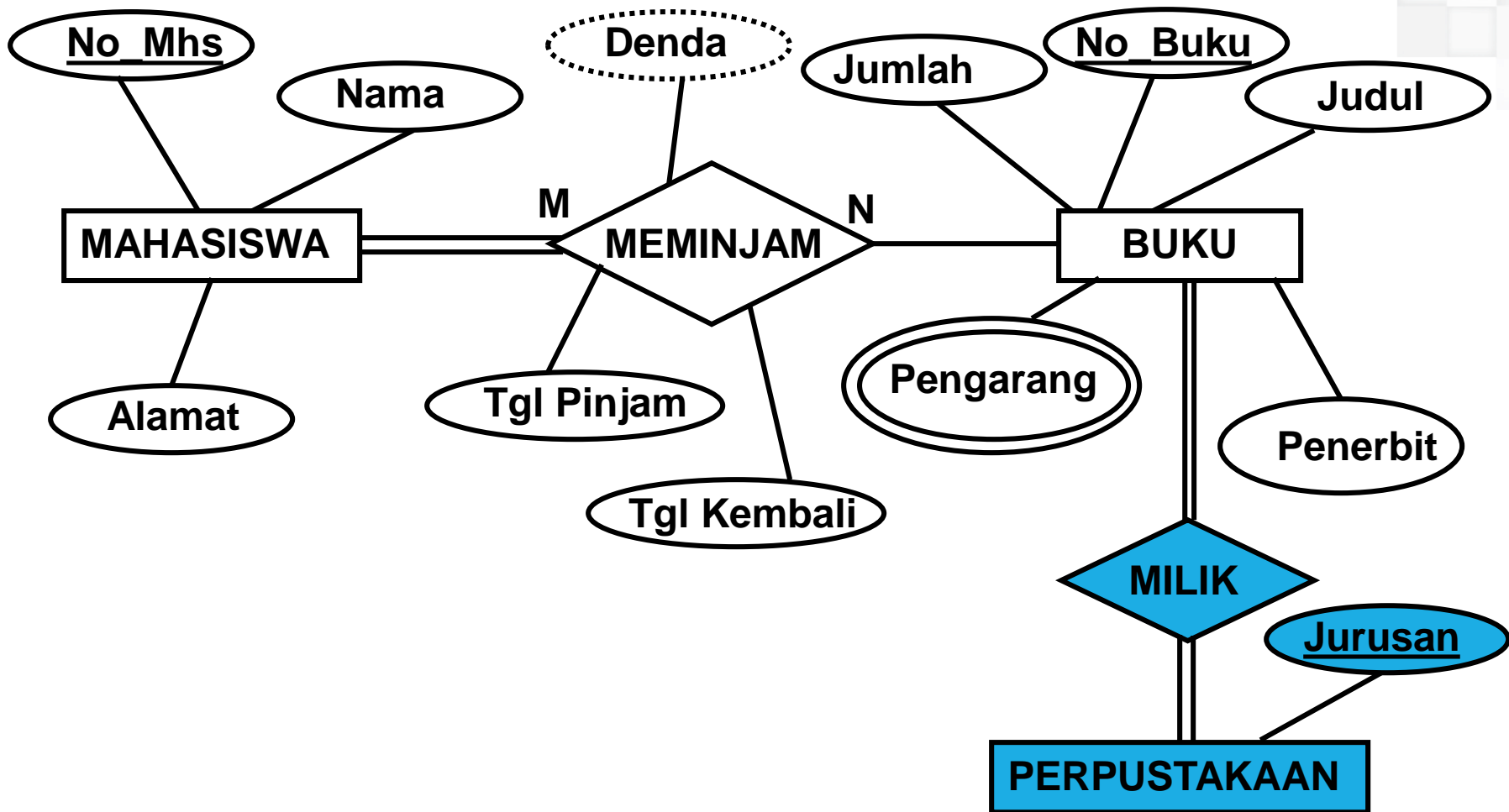


## Latihan – Soal 5

---

Seperti soal nomor 4, namun ada beberapa tambahan penjelasan berikut ini. Fakultas memiliki 3 jurusan, dan tiap jurusan memiliki beberapa perpustakaan. Mahasiswa boleh meminjam dari perpustakaan manapun di jurusan tersebut. Setiap perpustakaan memiliki banyak buku. Buku yang sama hanya ada di satu perpustakaan.

# Jawab: Latihan 5

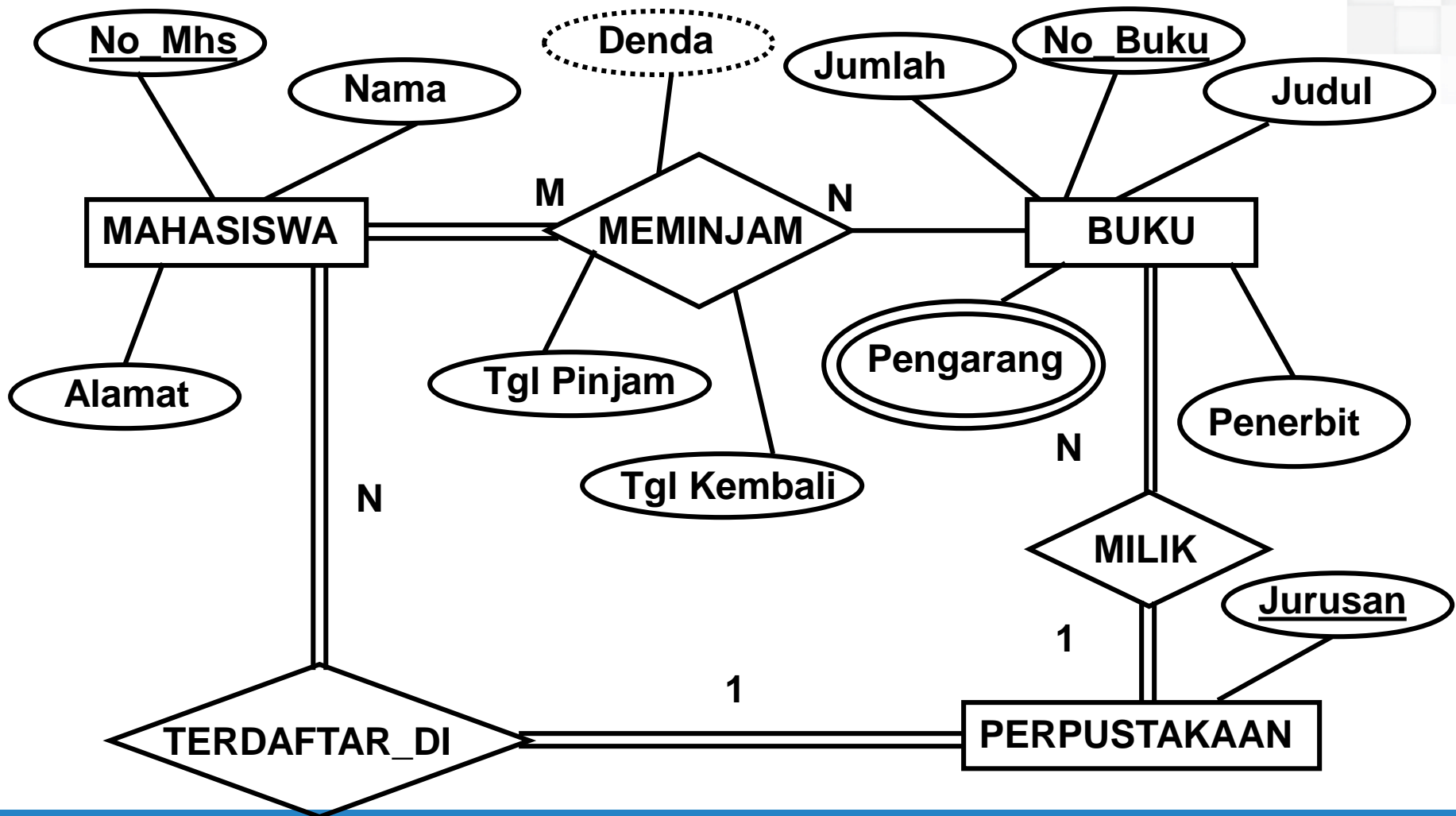


## Latihan – Soal 6

---

Seperti soal nomor 5, namun ada beberapa tambahan penjelasan berikut ini. Fakultas memiliki 3 jurusan, dan tiap jurusan memiliki perpustakaan. Setiap mahasiswa baru otomatis langsung terdaftar di satu perpustakaan, namun mereka boleh meminjam dari perpustakaan manapun di jurusan tersebut.

# Jawab: Latihan 6



# Latihan – Soal 7

---

For each airplane we keep track of its identity, number of seats, and type. Its type is defined by the name, the maximum number of seats, and the company that produces it. Specific airplane types can land at specific airports. For each airport we need to know the name, the city, the state, and the code of it. Each flight has information about the airline, the number and the weekdays it operates. It also has fares, with code number, amount, and restrictions. A flight has several flight legs (a flight leg is for example Aalborg-Copenhagen, and Copenhagen-New York, when you are flying from Aalborg to New York via Copenhagen), from and to an airport, with scheduled departure and arrival time respectively. Each airplane is assigned to a leg instance, which is a particular occurrence of a flight leg on a particular date, while a leg is a non-stop portion of a flight. Each leg instance arrives to and departs from airport at specific arrival and departure times. Finally, for each leg instance we can make reservations of seats, based on the customer name and customer phone number. Design an ER schema that accurately reflects the requirements.

# Latihan – Soal 8

Consider the following set of requirements for a university database that is used to keep track of students' transcripts.

- (a) The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birth date, sex, class (Stage 1, Stage 2, ..., Stage 4), major department, minor department (if any), and degree program (B.A., B.Sc., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student.
- (b) Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.
- (c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.
- (d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ...; up to the number of sections taught during each semester.
- (e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).