



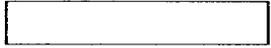
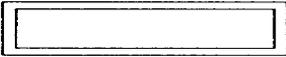
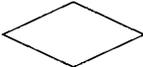
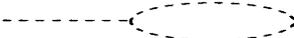
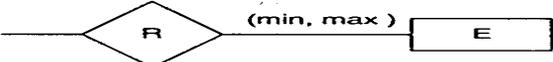
01

Review Basis Data 1

by: Ahmad Syauqi Ahsan

Notasi ER Diagram (El-Masri)

2

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

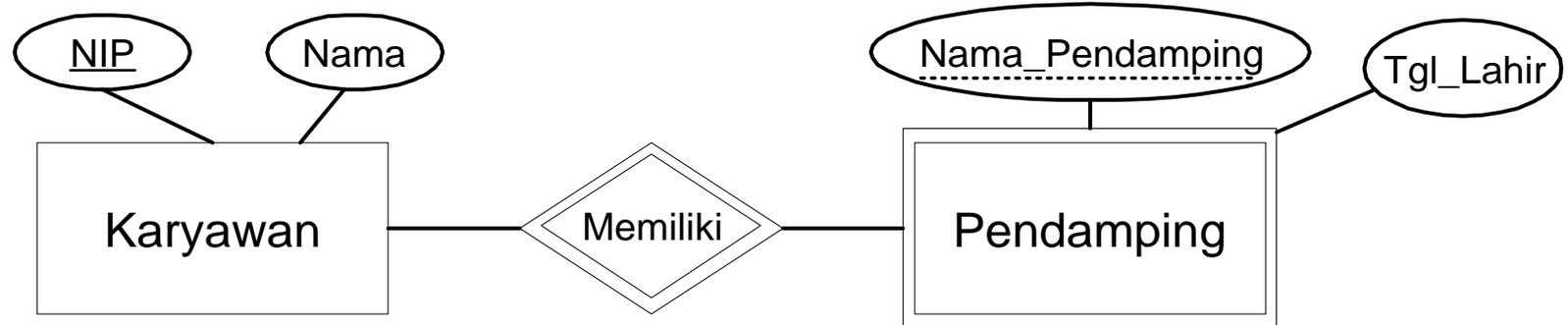
ER Diagram

- Sebuah database dapat dimodelkan sebagai:
 - ▣ Kumpulan dari beberapa entitas, dan
 - ▣ Relasi diantara entitas-entitas tersebut
- **Entitas** adalah suatu objek yang ada dan dapat dibedakan dari objek-objek yang lain.
 - ▣ Contoh: mahasiswa, fakultas, mata kuliah, dan lain-lain.
- Suatu Entitas mempunyai atribut:
 - ▣ Contoh: mahasiswa mempunyai nama, no telpon, alamat, dan lain-lain.
- Suatu Entitas dapat bersifat konkret/nyata (seperti: mahasiswa, dosen, dll) maupun bersifat abstrak (seperti: mata kuliah, jenis pekerjaan, dll).

Entitas Lemah

4

- Entitas lemah adalah suatu entitas yang keberadaannya sangat tergantung dengan entitas yang lain.
 - ▣ Tidak memiliki atribut kunci
 - ▣ Entitas tempat bergantung disebut identifying Owner.
 - ▣ Atribut dari entitas lemah berperan sebagai **Partial Identifier**.
- Contoh:



Tipe-tipe atribut

- **Simple/Atomic attribute:** adalah atribut yang tidak dapat dibagi lagi menjadi atribut yang lebih rinci.
- **Composite attribute:** atribut yang terdiri dari beberapa atribut lain yang lebih rinci.
 - ▣ Contoh: atribut ALAMAT bisa mempunyai atribut JALAN, KOTA, dan KODE_POS.
- **Single-valued attribute:** atribut yang hanya memiliki satu nilai.
 - ▣ Contoh: atribut UMUR pada entitas PEGAWAI.
- **Multi-valued attribute:** atribut yang bisa memiliki lebih dari satu nilai.
 - ▣ Contoh: atribut HOBI pada entitas MAHASISWA.

Tipe-tipe atribut (2)

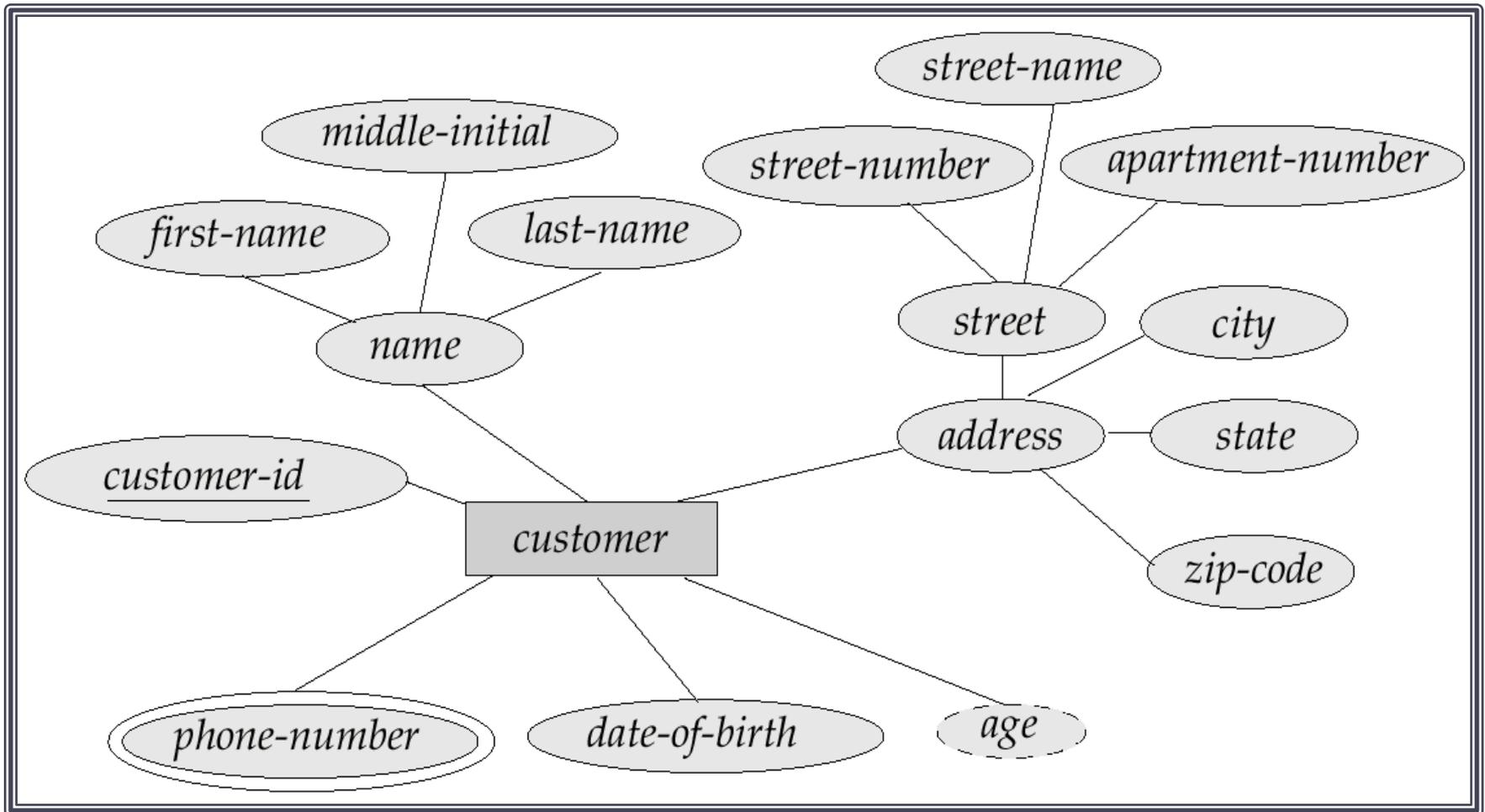
- **Derived attribute:** adalah atribut yang nilainya dapat diturunkan atau dihitung berdasarkan nilai dari atribut yang lain.

Contoh:

- ▣ Atribut UMUR dapat dihitung dari atribut TANGGAL_LAHIR.
- ▣ Atribut LAMA_KULIAH dapat dihitung dari atribut TANGGAL_MASUK

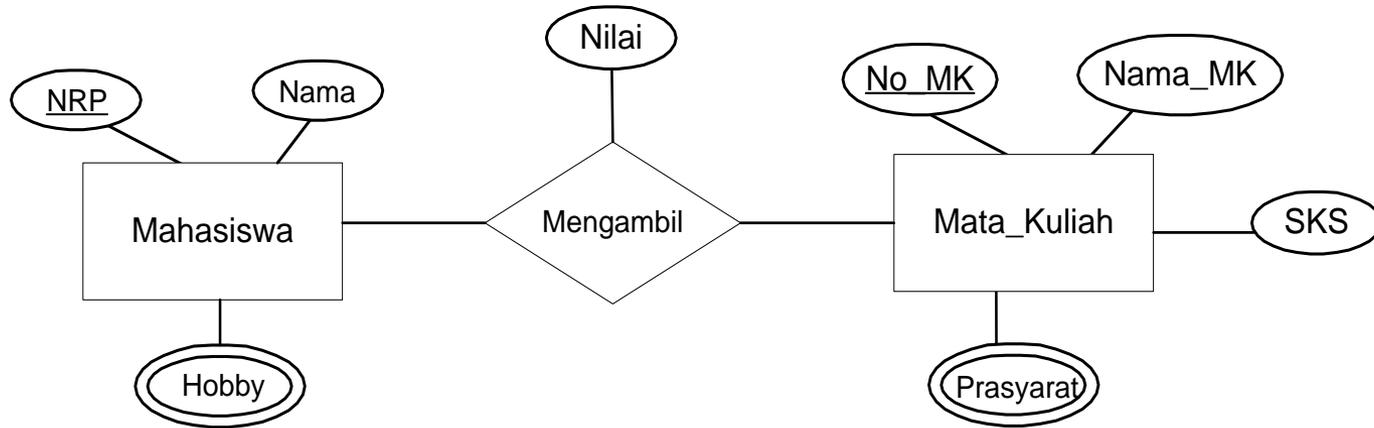
Contoh ER Diagram

7



Relasi dengan atribut

8



- Jika atribut NILAI diletakkan pada entitas MAHASISWA maka seorang mahasiswa akan mempunyai nilai yang sama untuk semua mata kuliah yang diambalnya.
- Jika atribut NILAI diletakkan pada entitas MATA_KULIAH, maka semua mahasiswa yang mengambil suatu mata kuliah akan mempunyai nilai yang sama.

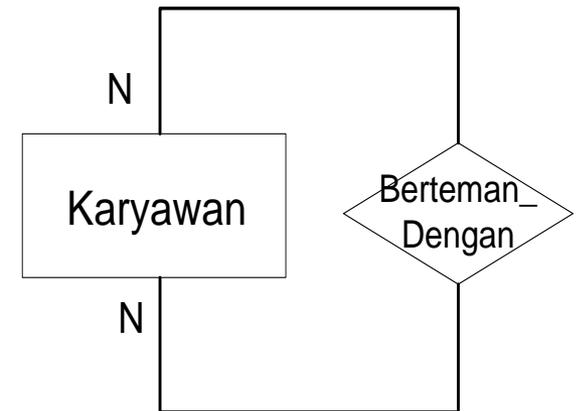
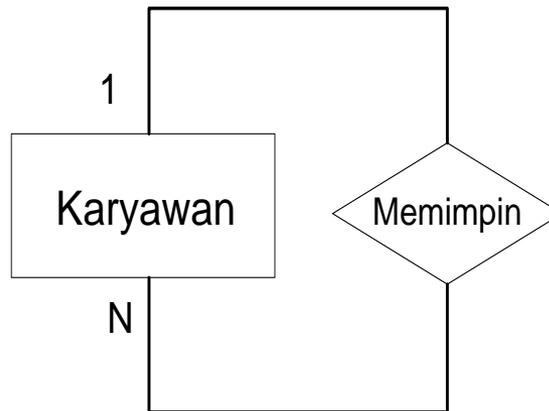
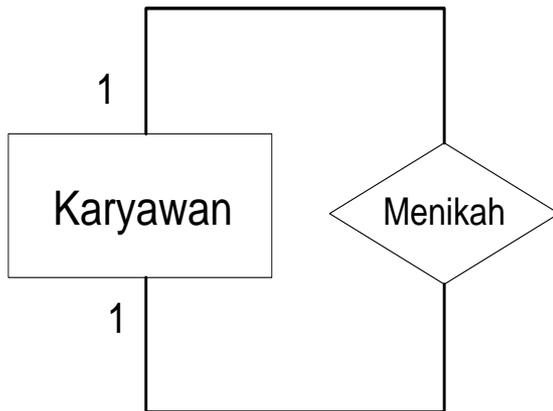
Derajat Relasi

- Derajat Relasi adalah jumlah entitas yang berpartisipasi dalam suatu relasi.
- Derajat Relasi dapat berupa:
 - Unary Relationship (Relasi berderajat 1)
 - Binary Relationship (Relasi berderajat 2)
 - Ternary Relationship (Relasi berderajat 3)

Unary Relationship

10

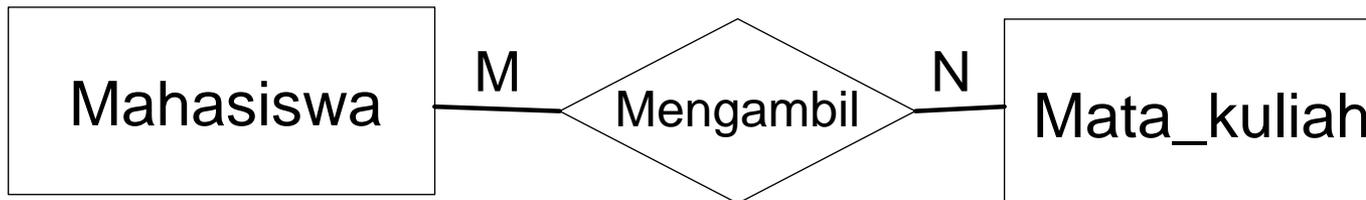
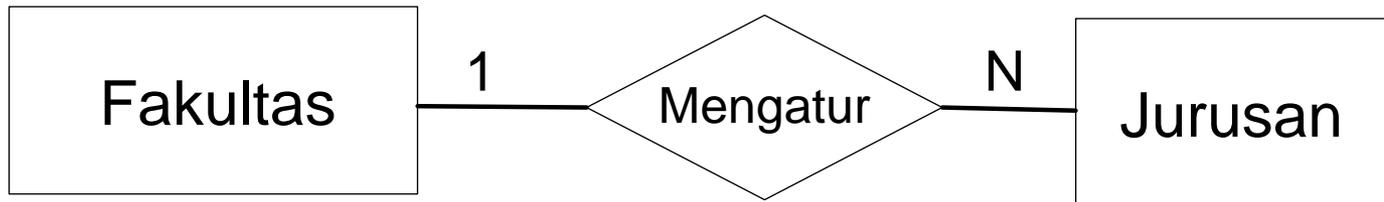
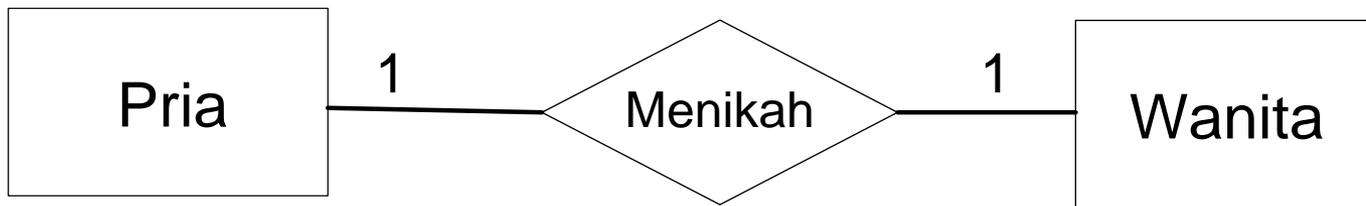
- Adalah relasi dimana jumlah entitas yang terlibat hanya satu
- Sering juga disebut sebagai recursive relationship (relasi rekursif)
- Contoh:



Binary Relationship

11

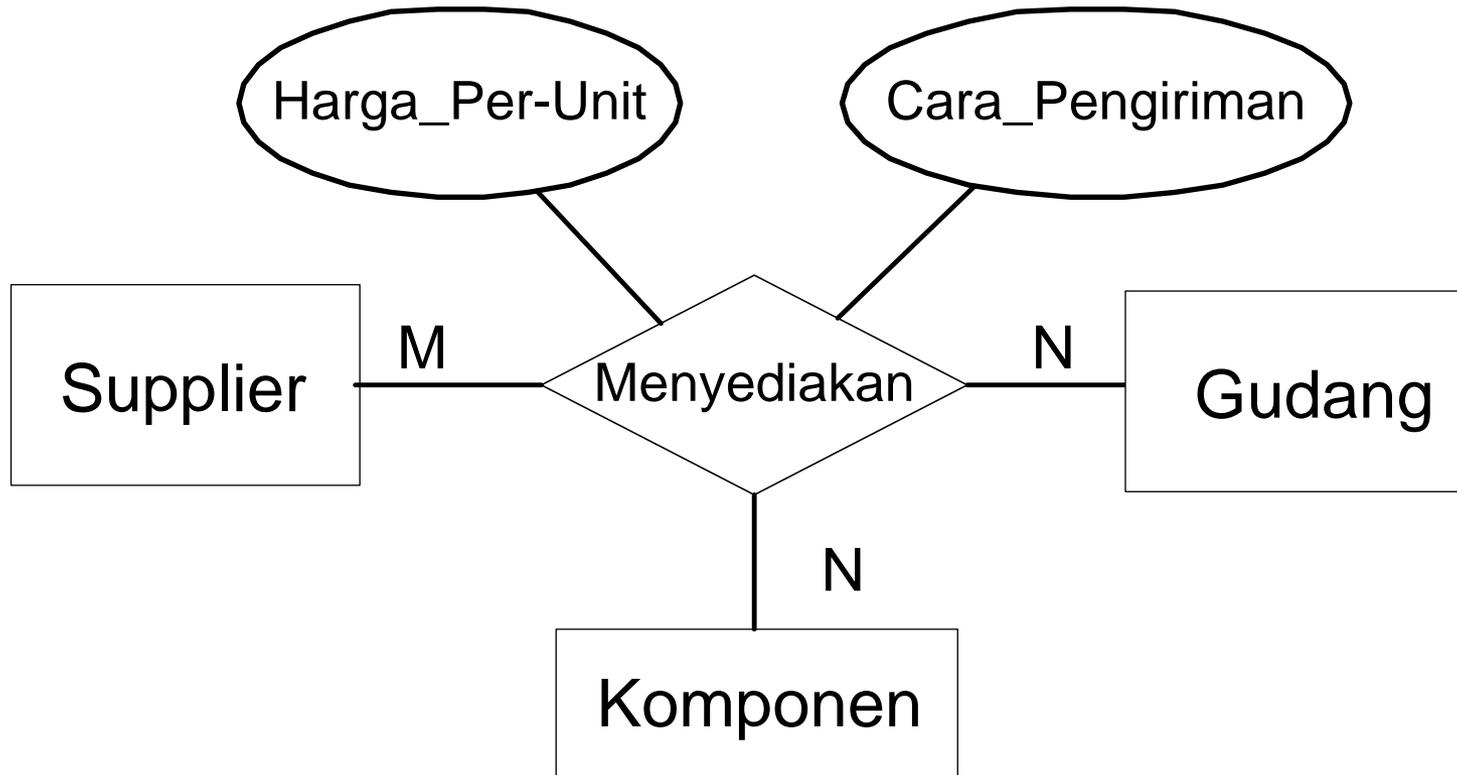
- Relasi yang melibatkan dua entitas
- Contoh:



Ternary Relationship

12

- Adalah relasi yang melibatkan 3 entitas.
- Contoh:

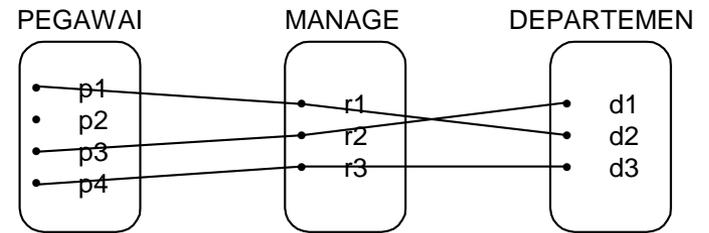


Rasio Kardinalitas

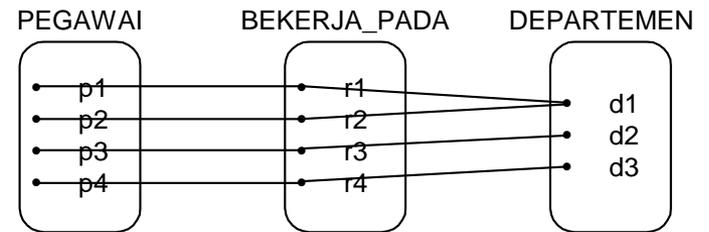
13

□ Rasio Kardinalitas dapat berupa:

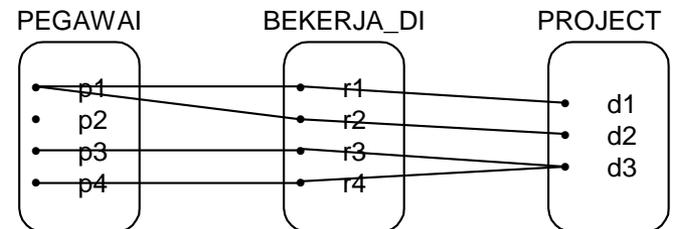
▣ One-to-one (1:1) relationship:



▣ One-to-many (1:M) relationship:



▣ Many-to-many (M:N) relationship:



Participation Constraint Dependency

14

- Digunakan untuk menunjukkan apakah keberadaan suatu entitas bergantung penuh atau tidak dengan entitas lainnya.
Ada 2 jenis Participation Constraint:
- Partisipasi total (**====**):
 - ▣ Adalah bentuk partisipasi yang menunjukkan ketergantungan penuh suatu entitas.
- Partisipasi parsial (**——**):
 - ▣ Adalah bentuk partisipasi menunjukkan ketergantungan tidak penuh suatu entitas



Mapping

15

- Mapping merupakan proses untuk menterjemahkan Conceptual Design Model (ER Diagram) ke bentuk Physical Design Model (skema relasi).
- Hasil dari proses Mapping ini dapat langsung digunakan untuk membuat tabel-tabel beserta constraintnya di dalam sistem database.

Langkah-langkah Mapping

16

1. Untuk setiap **Entitas Kuat**, buat skema relasi **R** yang menyertakan seluruh **Simple Attribute** dan **Simple Attribute** dari **Composite Attribute** yang ada, pilih **salah satu atribut kunci** sebagai **Primary Key**.
2. Untuk setiap **Entitas Lemah**, buatlah skema relasi **R** dengan mengikutsertakan seluruh **Simple Attribute**. Tambahkan **Primary Key** dari **entitas kuatnya (Owner Entity type)** yang akan digunakan sebagai **Primary Key** bersama-sama **Partial Key** dari **Entitas Lemah** (digabung).
3. Untuk setiap **relasi binary 1:1**, tambahkan Primary Key dari sisi yang lebih "**ringan**" ke sisi (entitas) yang lebih "**berat**". Suatu sisi dianggap lebih "**berat**" timbangannya apabila mempunyai **partisipasi total**. Tambahkan juga **Simple Attribute** yang terdapat pada relasi tersebut ke sisi yang lebih "**berat**".

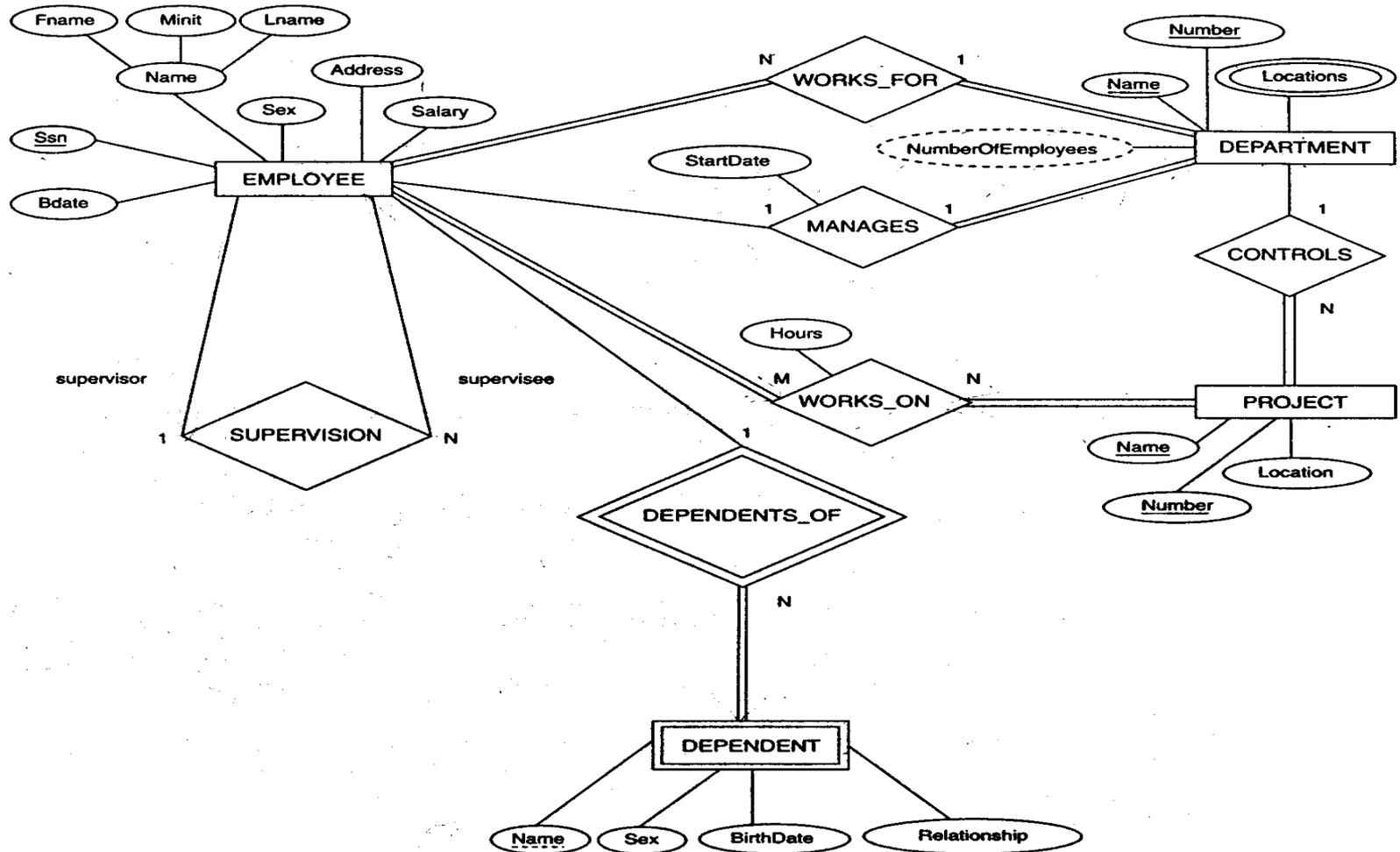
Langkah-langkah Mapping (2)

17

1. Untuk setiap relasi binary **1:N** yang **tidak melibatkan entitas lemah**, tentukan mana sisi yang lebih "**berat**". Sisi dianggap lebih "**berat**" timbangannya adalah **sisi-N**. Tambahkan **Primary Key** dari sisi yang "**ringan**" ke skema relasi sisi yang lebih "**berat**". Tambahkan juga **seluruh simple attribute** yang terdapat pada **relasi** biner tersebut.
2. Untuk setiap relasi **binary M:N**, **buatlah skema relasi baru R** dengan menyertakan **seluruh simple attribute** yang terdapat **pada relasi biner** tersebut. Tambahkan masing-masing **primary key** dari kedua sisi ke skema relasi **R** tersebut (sbg **foreign key**), lalu **digabung** menjadi **satu** membentuk **Primary Key** dari skema relasi **R**.
3. Untuk setiap **Multivalued Attribute**, buatlah skema relasi **R** yang menyertakan **atribut dari multivalued** tersebut. Tambahkan **Primary Key** dari relasi yang memiliki multivalued tersebut. Kedua atribut tersebut membentuk **Primary Key** dari skema relasi **R**.
4. Untuk setiap relasi **N-ary** dengan $N > 2$, buatlah skema relasi **R** yang menyertakan seluruh **Primary Key** dari entitas yang ikut serta. Sejumlah n **Foreign Key** tersebut akan membentuk **Primary Key** untuk skema relasi **R**. Tambahkan seluruh Simple Attribute yang terdapat pada relasi n -ary tersebut.

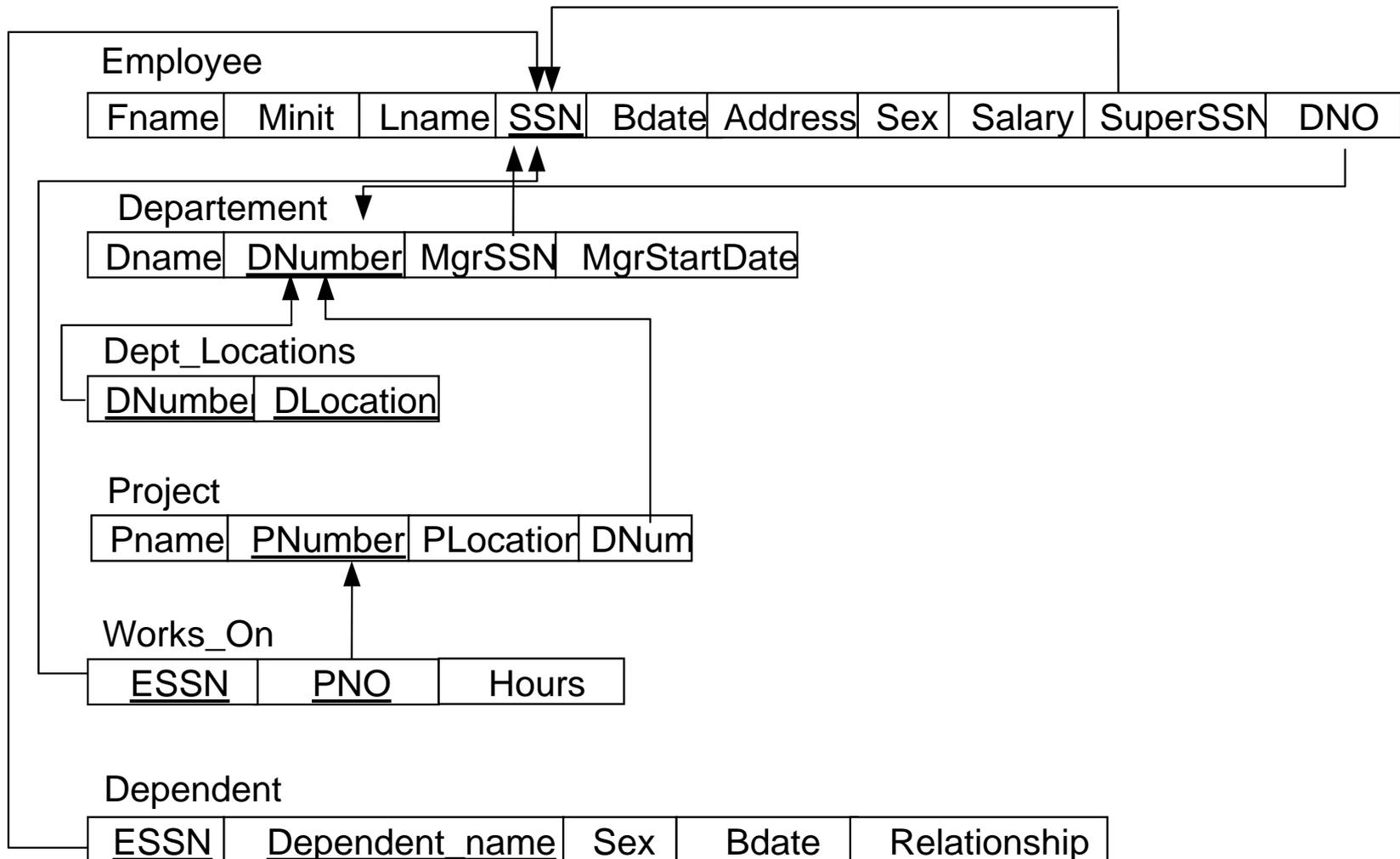
Contoh proses Mapping (CDM)

18



Contoh proses Mapping (PDM)

19



Normalisasi

20

- Normalisasi merupakan proses perbaikan skema database yang bertujuan untuk mengurangi/menghilangkan adanya informasi yang disimpan secara redundant.
- Dengan membuat skema database yang sudah di-normal-kan, akan mempermudah proses modifikasi dari skema database tersebut di waktu yang akan datang.
- Bentuk normalisasi yang sering digunakan adalah **1st NF**, **2nd NF**, **3rd NF**, dan **BCNF**.
- Normalisasi dengan derajat lebih tinggi akan lebih baik dari normalisasi dengan derajat dibawahnya.
- Tidak selamanya harus menggunakan normalisasi dengan level tertinggi. Pada banyak kasus, bentuk **3rd NF** sudah cukup

1st NF

21

- Suatu relation R disebut **1st NF** jika dan hanya jika semua attribute value-nya simple/atomic (tidak boleh ada attribute yang **composit & multivalued**)
- Tujuan **1st NF** adalah:
 - ▣ Membuang adanya pengulangan (**Redudansi**) data,
 - ▣ Menghindari adanya pencatatan **Null Value**, dan
 - ▣ Menjaga setiap entry data dr relasi (perpotongan baris-kolom) memiliki **maksimal satu nilai tunggal**.

2nd NF

22

Suatu relation R disebut dalam bentuk 2nd NF jika:

- Sudah dalam bentuk 1st NF
- Setiap atribut bukan kunci (bukan PK) bergantung penuh (FULL DEPENDENT) pada kunci primer (PK).
- Sehingga **tidak ada atribut bukan kunci** yang bergantung pada sebagian (parsial) kunci primer.

3rd NF

23

Suatu relation R disebut dalam bentuk 3rd NF jika:

- Sudah dalam bentuk **2nd NF**
- Setiap atribut bukan kunci tidak memiliki ketergantungan transitif terhadap kunci primer.
- **Kebergantungan transitif (transitive dependency)** adalah ketergantungan fungsional antara 2 (atau lebih) atribut **bukan key** (kunci/PK).

Tanya Jawab

Terima Kasih