

Model EER

(Enhanced Entity Relationship)

Model EER :

Model Entity Relationship yang ditambah kemampuan semantiknya dengan beberapa konsep yang lebih kompleks.

Konsep-konsepnya yaitu :

- Subclass
- Superclass
- Generalization
- Spezialization
 - Specialization Hierarchy
 - Spezialization Lattice
- Attribut Inheritance
- Shared subclass
- Categorization

Sub-Kelas (Subclass) :

- Subset dari suatu entitas yang dikelompokkan dalam pengertian tertentu yang perlu disajikan secara eksplisit.
- Contoh : Entitas PEGAWAI mempunyai beberapa *subclass* seperti : SEKRETARIS, TEKNISI, AHLI (gb.1)

Super-kelas (Superclass) :

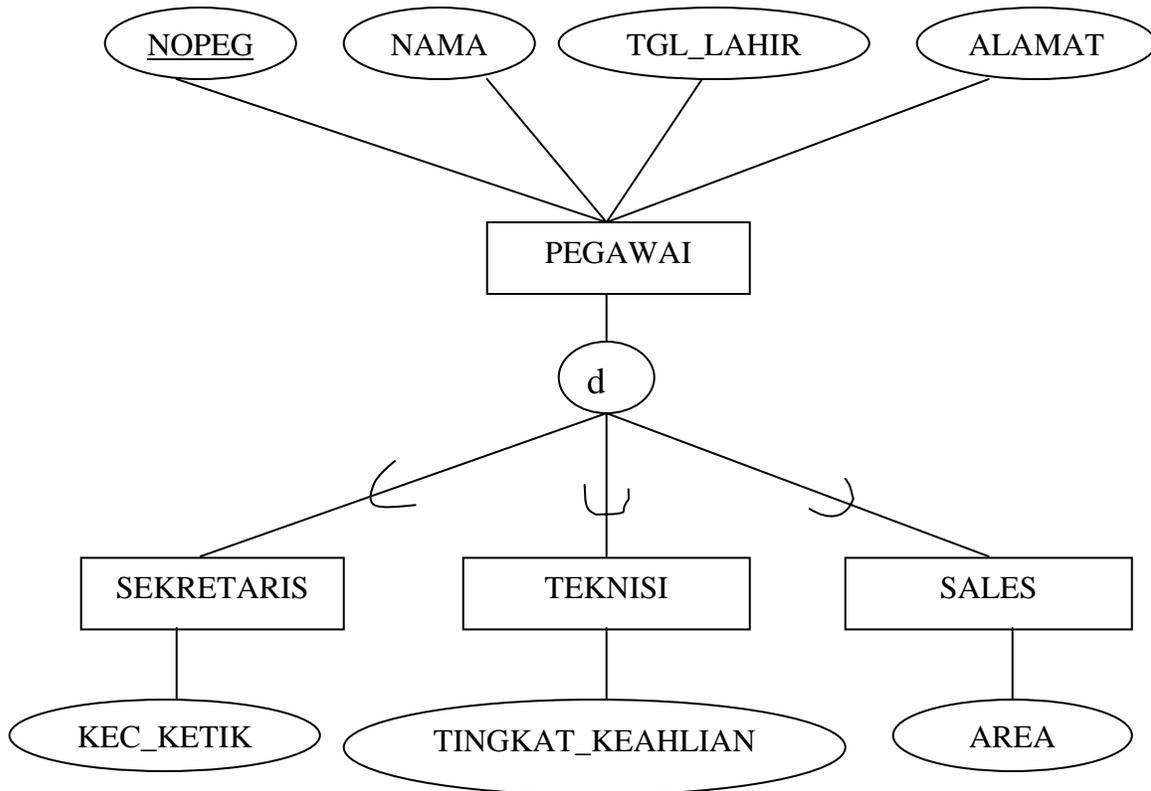
- Entitas yang merupakan induk dari subclass-subclassnya.
- Contoh : Subclass SEKRETARIS, TEKNISI, SALES mempunyai *superclass* PEGAWAI (gb.1)

Spesialisasi (Spezialization) :

- Proses pemecahan entitas menjadi subclass-subclass beserta atribut-atributnya.
- Terdapat beberapa jenis spesialisasi seperti Disjoint Total, Disjoint Partial, Overlapping Total dan Overlapping Partial.

- Contoh :

Spesialisasi dari PEGAWAI berdasarkan tipe pekerjaan

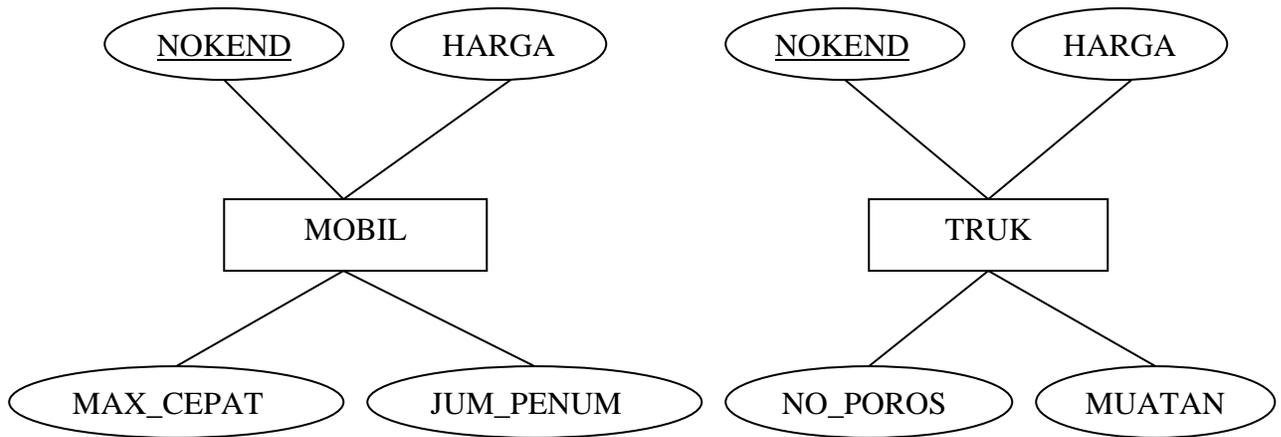


Gb. 1
Spesialisasi tipe pekerjaan entity PEGAWAI

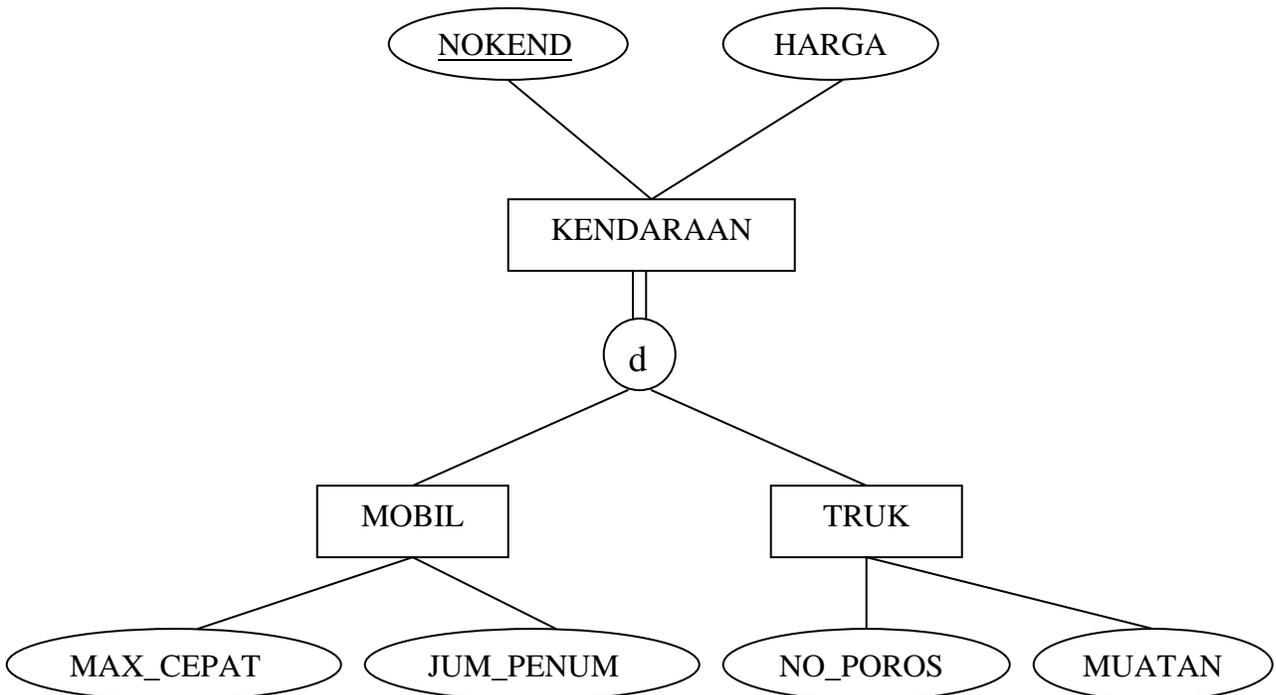
Generalisasi (Generalization) :

- Proses penggabungan subclass-subclass menjadi suatu entitas yang lebih umum.

Contoh : Generalisasi dari MOBIL dan TRUK menjadi KENDARAAN



Gb. 2
Tipe entity Mobil & Truk



Gb. 3
Generalisasi dari Mobil dan Truk menjadi Kendaraan

Subclass / Superclass Relationship :

- Relationship yang menghubungkan subclass dan superclassnya.
- Bila suatu entity yang merupakan anggota dari suatu subclass, maka ia juga merupakan anggota dari superclass.
- Bila suatu entity merupakan anggota dari suatu superclass maka ia belum tentu merupakan anggota dari subclass.

Attribut Inheritance :

- Merupakan pewarisan atribut dari superclass – nya
- Entity yang merupakan anggota dari subclass mewarisi semua atribut dari Entity superclass – nya.

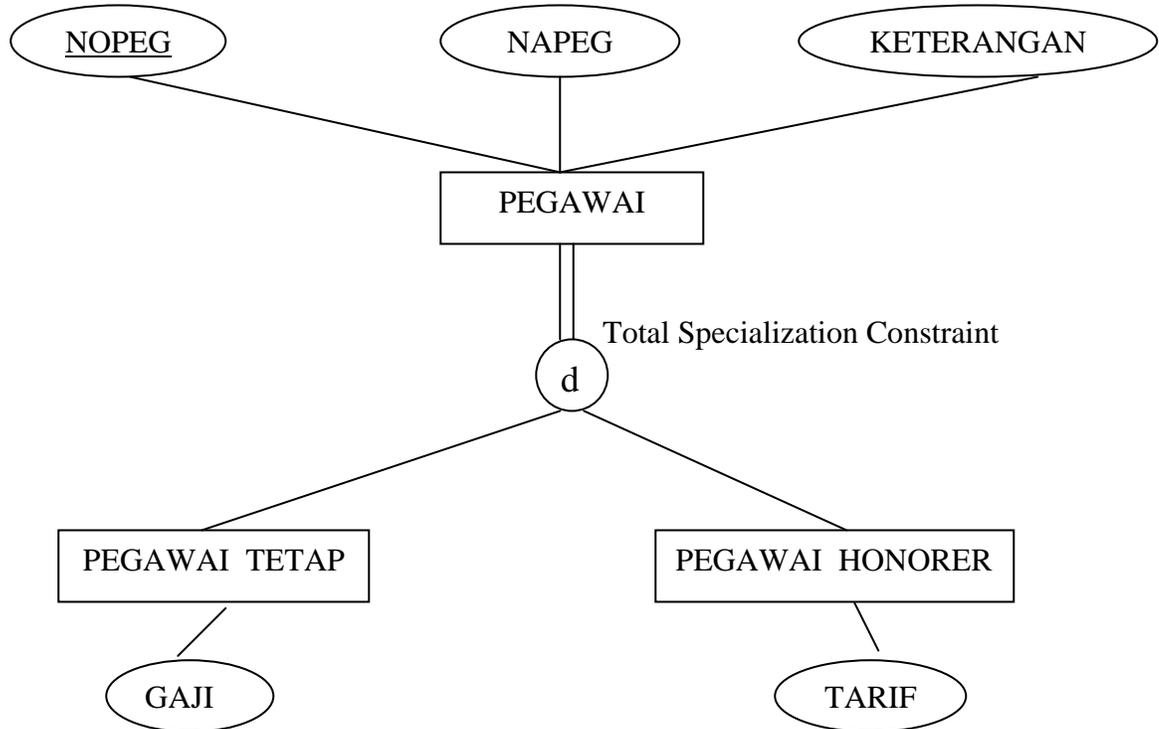
Disjoint Constraint :

- Constraint yang menerangkan bahwa subclass-subclass dari spesialisasi saling disjoint, artinya entity merupakan anggota dari salah satu subclass.
- Disjoint Constraint direpresentasikan dengan lambang "d" yang berarti *disjoint*.

Contoh : entity dari spesialisasi tipe pekerjaan dari PEGAWAI

merupakan anggota dari subclass :

PEGAWAI TETAP atau PEGAWAI HONORER.



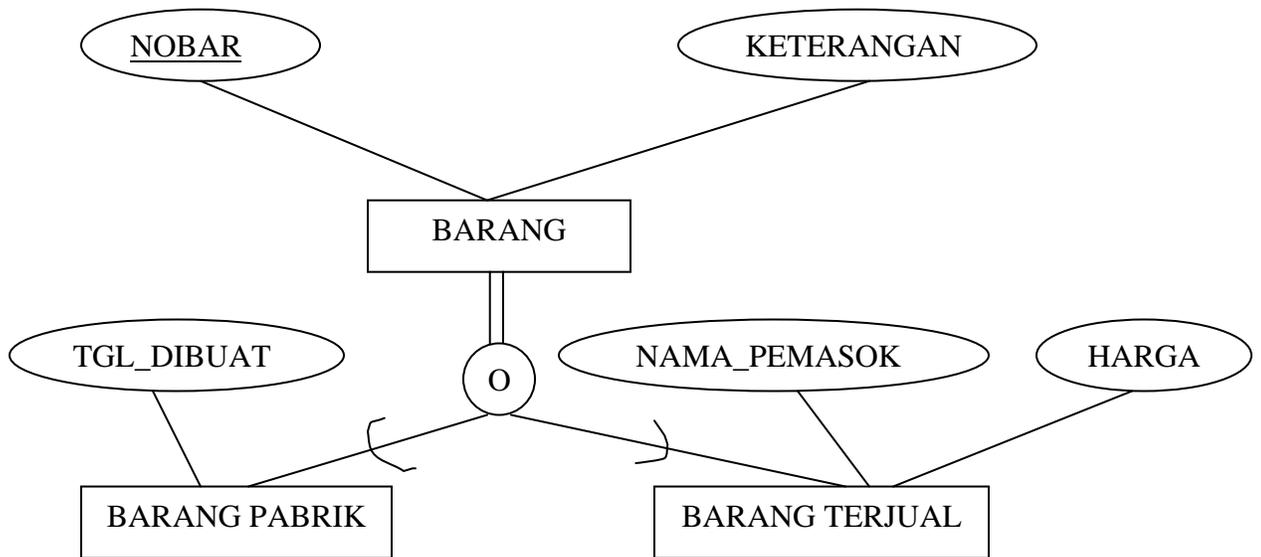
Gb. 4
Disjoint Constraint

Non – Disjoint Constraint :

- Constraint yang menerangkan bahwa subclass-subclass dari spesialisasi tidak saling disjoint, artinya entity mungkin anggota lebih dari satu subclass.

- Non-Disjoint digambarkan dengan lambang "o" yang berarti *overlapping*.

Contoh : Entity dari spesialisasi tipe barang merupakan anggota dari subclass BARANG PABRIK dan juga anggota dari subclass BARANG TERJUAL.



Gb. 5
Non-Disjoint Constraint

Total Specialization Constraint :

- Constraint yang menerangkan bahwa setiap entity di dalam superclass harus merupakan anggota dari salah satu subclass.

Contoh : Entity PEGAWAI harus termasuk subclass dari

PEGAWAI TETAP atau PEGAWAI HONORER.

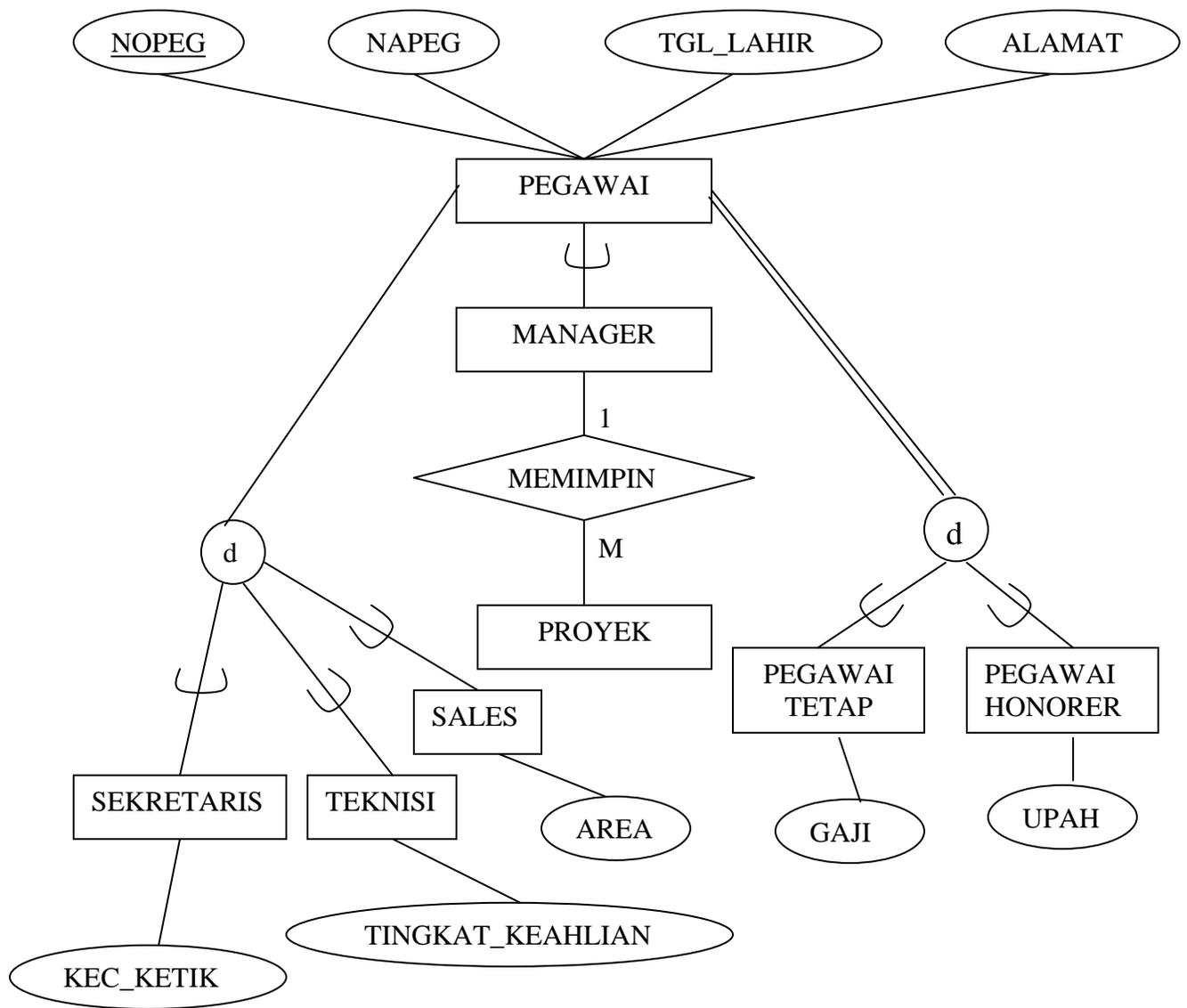
(Gb. 4)

Partial Specialization Constraint :

- Constraint yang menerangkan bahwa setiap entity didalam superclass dapat merupakan anggota dari subclass-subclass yang didefinisikan.

Contoh : Entity dari PEGAWAI dapat merupakan anggota

dari subclass MANAGER, TEKNISI atau SALES.



Gb. 6

Spesialisasi berdasarkan tipe pekerjaan dan kontrak dari pegawai / metode pembayaran

Specialization Hierarchy :

Spesialisasi bertingkat dimana setiap subclass berpartisipasi didalam satu kelas / subclass relationship

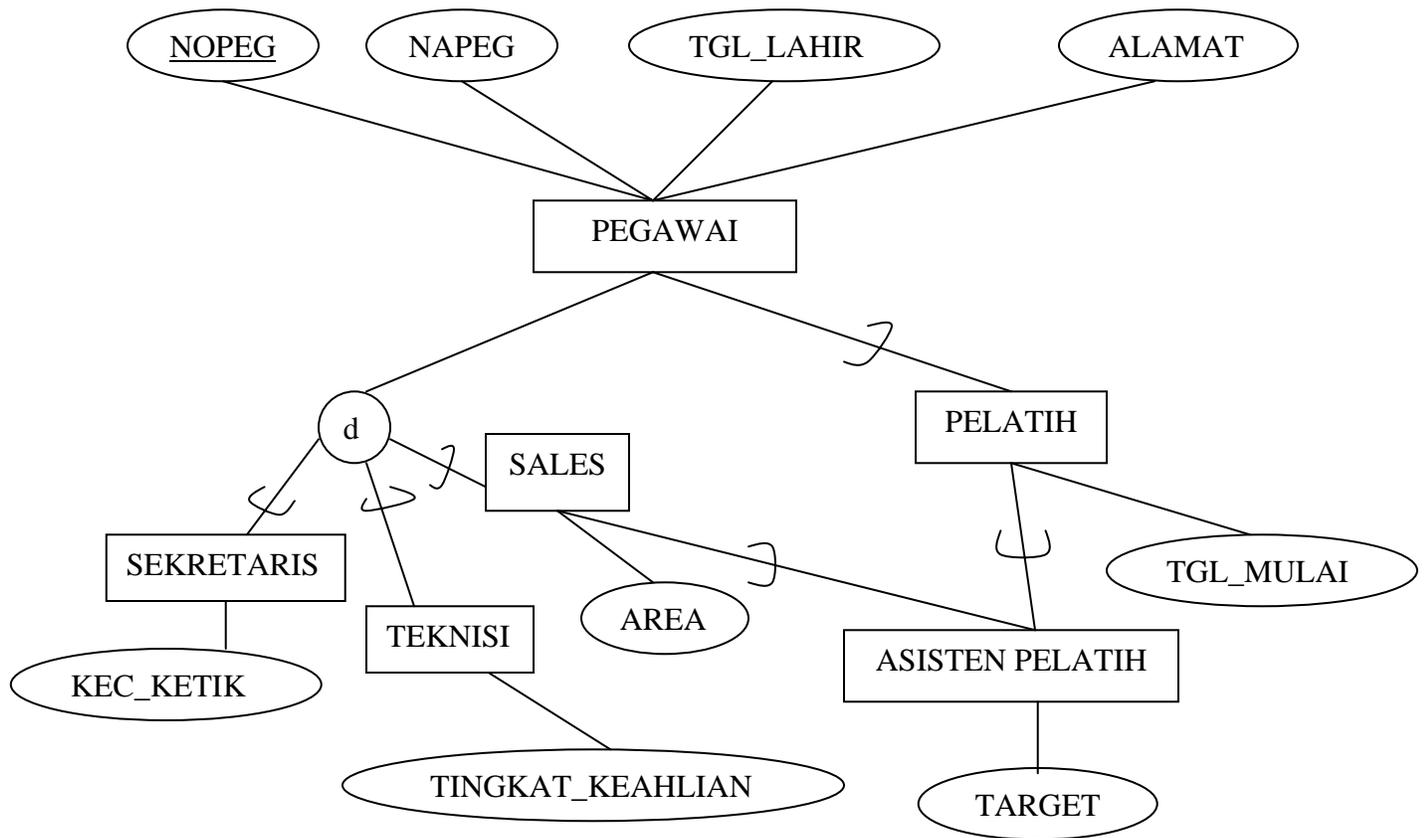
- **Specialization Lattice :**

Spesialisasi bertingkat dimana suatu subclass dapat berpartisipasi didalam beberapa class / subclass relationship.

- **Shared – Subclass :**

Subclass yang mempunyai lebih dari satu superclass.

Contoh : Subclass ASISTEN PELATIH mempunyai dua superclass yang tipenya sama yaitu SALES & PELATIH.



Gb. 7

Spesialisasi dengan subclass yang digunakan bersama
(Shared subclass)

Kategori :

adalah kebutuhan yang timbul untuk model suatu relationship superclass/subclass tunggal dengan lebih dari satu superclass dimana superclass-superclass tersebut menggambarkan jenis entity yang berbeda. Sebuah kategori mempunyai satu atau lebih superclass yang dapat mewakili tiap tipe entity, dimana superclass/subclass lainnya bisa saja hanya mempunyai satu superclass.

Kategori dapat saja untuk seluruh relasi atau hanya sebagian saja.

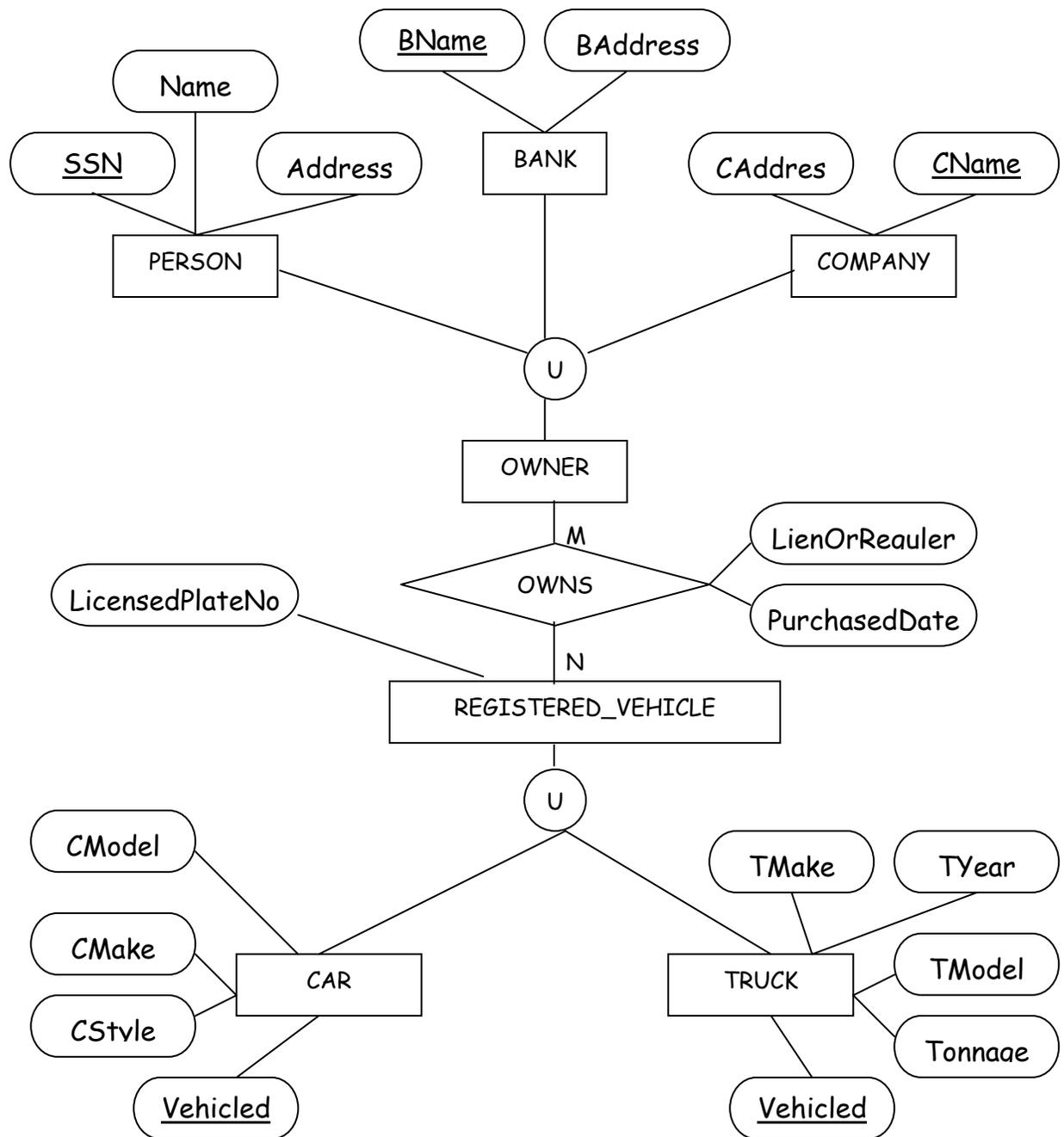
Contoh :

Terdapat 3 jenis entitas yaitu : PERSON, BANK dan COMPANYY.

Dalam suatu database REGISTERED_VEHICLE, pemilik kendaraan (OWNER) bisa saja : perorangan, bank atau perusahaan.

Kita perlu membuat suatu class yang terdiri-dari 3 jenis entitas untuk memainkan perannya sebagai pemilik kendaraan.

Maka dibuat suatu kategori OWNER yaitu sebuah subclass dari gabungan (UNION) 3 class : COMPANYY, BANK, PERSON

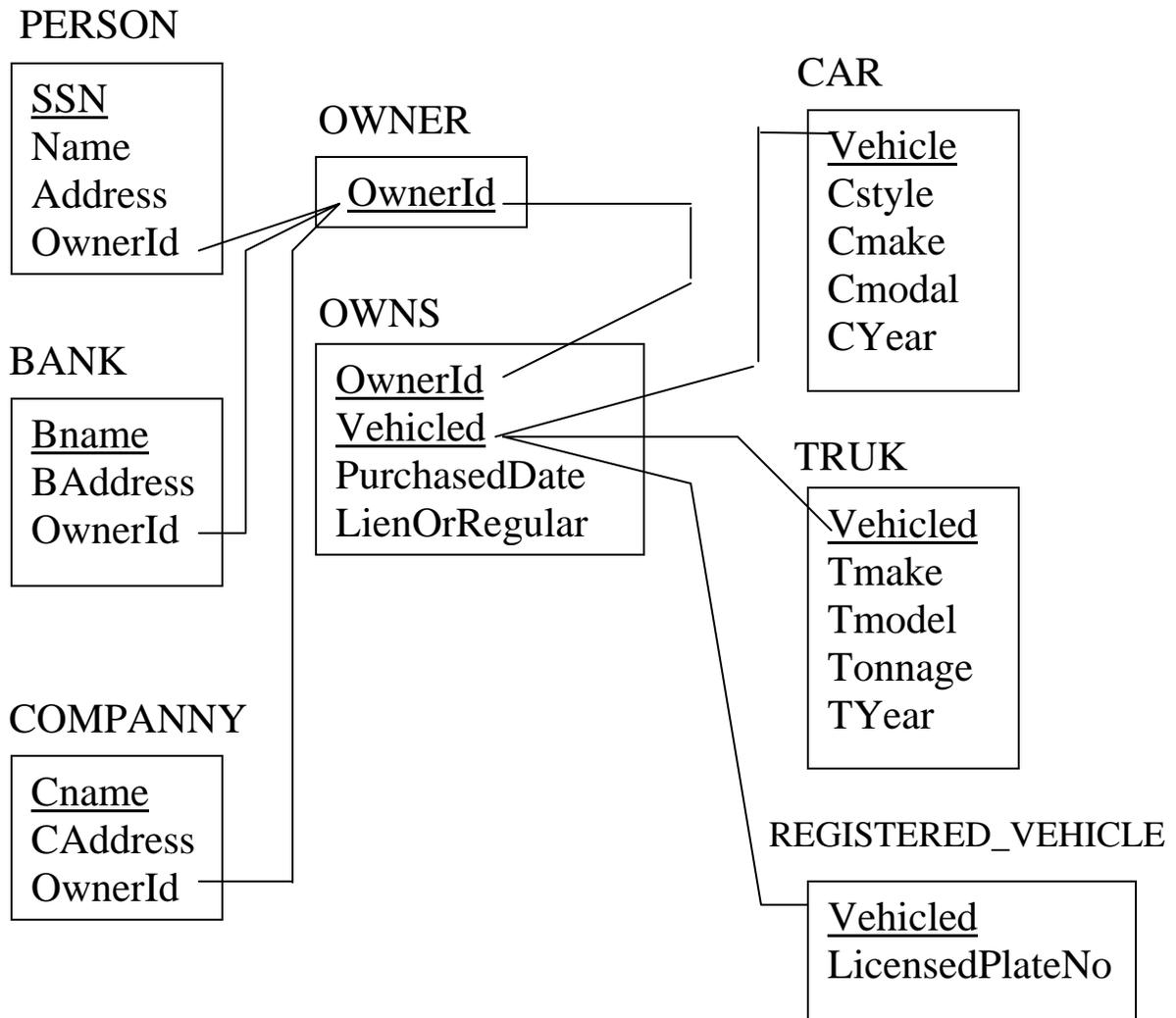


Keterangan :

Pada gambar di atas, terdapat 2 kategori yaitu :

1. OWNER yang merupakan sebuah subclass dari gabungan PERSON, BANK dan COMPANY
2. REGISTERED_VEHICLE yang merupakan subclass dari gabungan CAR dan TRUCK

Pemetaan EER menjadi model relasional :



Pemetaan (mapping) EER ke model Relasional

Algoritma pemetaannya merupakan perluasan atau lanjutan dari pemetaan ER.

Langkah 13 : Untuk memetakan Spesialisasi.

$\text{Attrs}(R)$ = atribut-atribut relasi R , sedangkan $\text{PK}(R)$ = primary key pada R . Untuk setiap spesialisasi dengan m subclass $\{S_1, S_2, \dots, S_m\}$ dan superclass C , dimana atribut C adalah $\{k, a_1, a_2, \dots, a_n\}$ dan $k = \text{PK}$.

Ini dapat dipetakan dengan beberapa cara berikut.

- a. Buat suatu relasi L dengan atribut-atribut $\text{Attrs}(L) = \{k, a_1, a_2, \dots, a_n\}$ dan $\text{PK}(L) = k$.

Buat juga relasi L_i untuk setiap subclass S_i , $1 \leq i \leq m$, dengan atribut

$\text{Attrs}(L_i) = \{k\} \cup \{\text{attrs } S_i\}$ dan $\text{PK}(L_i) = k$.

Perhatikan gb. 1, mappingnya :

PEGAWAI

<u>NOPEG</u>	NAMA	ALAMAT	TGL_LAHIR
--------------	------	--------	-----------

SEKRETARIS

<u>NOPEG</u>	KEC_KETIK
--------------	-----------

TEKNISI

<u>NOPEG</u>	TINGKAT_KEAHLIAN
--------------	------------------

SALES

<u>NOPEG</u>	AREA
--------------	------

b. Buat relasi L_i untuk setiap subclass S_i , $1 \leq i \leq m$ dengan atribut-atribut

$$\text{Attrs}(L_i) = \{\text{Attrs } S_i\} \cup \{k, a_1, a_2, \dots, a_n\} \text{ dan } \text{PK}(L_i) = k.$$

Perhatikan gb. 3 , mappingnya :

MOBIL

<u>NOKEND</u>	HARGA	MAX_CEPAT	JUM_PENUM
---------------	-------	-----------	-----------

TRUK

<u>NOKEND</u>	HARGA	NO_POROS	MUATAN
---------------	-------	----------	--------

- c. Buat satu relasi L dengan atribut $\text{Attrs}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{attributes of } S_i\} \cup \dots \cup \{\text{attributes dari } S_m\} \cup \{t\}$ dan $\text{PK}(L)=k$.
t adalah atribut-atribut yang dimiliki subclass.

Perhatikan gb. 1, mappingnya :

PEGAWAI

<u>NOPEG</u>	NAMA	ALAMAT	TGL_LAHIR	KEC_KETIK	...
...	TINGKAT_KEAHLIAN	AREA			

- d. Buat satu skema relasi L dengan $\text{Attrs}(L) = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{attributes of } S_i\} \cup \dots \cup \{\text{attributes dari } S_m\} \cup \{t_1, t_2, \dots, t_m\}$ dan $\text{PK}(L)=k$. Setiap $t_i, 1 \leq i \leq m$ adalah atribut Boolean (flag) yang menunjukkan bukan tuple yang dimiliki oleh subclass S_i .

Perhatikan gb. 5, mappingnya :

BARANG

<u>NOBAR</u>	KETERANGAN	P_FLAG	TGL_DIBUAT	...
...	J_FLAG	NAMA_PEMASOK	HARGA	

Transformasi dari EER ke Database Relasional

Pilihan 1.

Buat relasi dari Super-Kelasnya sendiri ditambah dengan relasi-relasi sebanyak Sub-Kelasnya, dimana relasi untuk tiap Sub-Kelasnya berisi atribut dari masing-masing Sub_Kelas ditambahkan Primary Key dari Super-Kelasnya sebagai Primary Keynya.

Pilihan 2.

Dibuat relasi-relasi sebanyak Sub-Kelasnya, dimana isi dari relasi tersebut adalah atribut-atribut dari Super-Kelas ditambahkan atribut-atribut dari Sub-Kelasnya sendiri. Primary key menggunakan Primary key Super-Kelasnya.

Disjoint

Dibuat satu relasi yaitu gabungan atribut Super-Kelasnya dengan atribut-atribut Sub-Kelasnya.

overlapping

Dibuat satu relasi yaitu gabungan atribut-atribut Super-Kelasnya dengan atribut-atribut Sub_Kelasnya, tetapi ditambahkan Flag sebanyak Sub-Kelasnya.