

BASIS DATA

MODEL RELASIONAL

MODEL RELASIONAL

- Diusulkan oleh Codd pada tahun 1970
- Model saat itu adalah hierarki dan jaringan
- Saat ini model relasional menjadi model data yang dominan dan mendasar pada produk andalan SMDB
- Digunakan luas oleh berbagai vendor seperti Informix, Oracle, Sybase, Access, SQL Server, FoxBase, Paradox, ...

MODEL RELASIONAL

- Database merupakan kumpulan dari satu atau lebih relasi.
- Masing – masing relasi merupakan tabel yang terdiri dari baris dan kolom.
- Keuntungannya jika dibandingkan dengan model data sebelumnya adalah representasi datanya sederhana dan mudah diekspresikan dalam bentuk query.

TERMINOLOGI MODEL RELASIONAL

- Sebuah baris disebut sebuah ***tuple/ row / record***
- Suatu header dari kolom disebut ***attribute/ field***
- Tipe data yang menjelaskan jenis-jenis nilai yang dapat muncul dalam setiap kolom disebut ***domain***
- Domain menjelaskan tipe data, menyangkut nama, format data, dan informasi tambahan untuk memudahkan menginterpretasikan nilai-nilai domain

Contoh:

Domain “Umur pegawai” : adalah pegawai suatu perusahaan, masing-masing misalnya harus mempunyai nilai umur antara 16 dan 50 tahun.

PENGANTAR MODEL RELASIONAL

- Kontruksi utama merepresentasikan data dalam model relasional adalah relasi.
- Relasi terdiri dari skema relasi dan contoh relasi.
- Skema relasi
 - Mendeskripsikan kepala kolom/field dari tabel.
- Contoh Relasi : tabel

Skema Relasi → Contoh Relasi

PENGANTAR MODEL RELASIONAL

Skema Relasi :

- Menentukan nama relasi, nama masing-masing field (kolom/atribut) dan domain dari masing-masing field.
- Domain yang diacu memiliki nama domain dan kumpulan nilai yang sesuai.

Contoh :

- Students(sid : **string**, name: **string**, login : **string**, age : **integer**, gpa : **real**)

PENGANTAR MODEL RELASIONAL

sid	name	login	age	gpa
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

PENGANTAR MODEL RELASIONAL

- Skema relasi menentukan domain untuk tiap field.
- Domain pada tiap field pada dasarnya adalah tipe dari field (bhs pemrograman).
- Secara formal :

$R(f_1:D_1, \dots, f_n:D_n)$ skema relasi untuk tiap f_i , di mana $1 \leq i \leq n$

$\{ \langle f_1:D_1, \dots, f_n:D_n \rangle \mid D_i \text{ anggota } \text{Dom}_i \}$

PENGANTAR MODEL RELASIONAL

- Relasi dengan menggunakan SQL
SQL menggunakan kata tabel untuk menyatakan relasi.
- Membuat tabel

```
CREATE TABLE students(sid char(20),  
                        name char(30),  
                        login char(20),  
                        age integer,  
                        gpa real)
```

PENGANTAR MODEL RELASIONAL

- Menyisipkan tuple

```
INSERT INTO students(sid, name, login, age, gpa)  
VALUES ('53688','Smith','smith@ee',18,3.2)
```

- Menghapus tuple

```
DELETE FROM students S  
WHERE S.name='Smith'
```

PENGANTAR MODEL RELASIONAL

Mengubah nilai-nilai kolom dalam baris

```
UPDATE students S
```

```
SET S.age=S.age+1,S.gpa=S.gpa-1
```

```
WHERE S.sid='53688'
```

Atau

```
UPDATE students S
```

```
SET S.gpa=S.gpa-1
```

```
WHERE S.gpa >=3.3
```

BATASAN INTEGRITAS

Kondisi yang ditetapkan pada skema database relasional dan membatasi data yang dapat disimpan dalam contoh database relasional.

Macam-macam Batasan Integritas

- Batasan Kunci
- Batasan Kunci Tamu
- Batasan Umum

BATASAN INTEGRITAS

Batasan Kunci

- Subset minimal tertentu pada field sebuah relasi merupakan identifier yang unik untuk suatu tuple.
- Candidate-key, merupakan himpunan atribut minimal yang dapat membedakan setiap baris data dengan unik dalam sebuah tabel.

BATASAN INTEGRITAS

- Batasan Kunci

```
CREATE TABLE Students(sid CHAR(20),  
                        name CHAR(30),  
                        login CHAR(20),  
                        age INTEGER,  
                        gpa REAL,  
                        UNIQUE(name,age),  
                        CONSTRAINT StudentsKey PRIMARY KEY(sid))
```

- Batasan Kunci Tamu

Misal terdapat relasi lain (Enrolled)

Enrolled(sid:string, cid:string, grade:string)

Kunci Tamu

cid	grade	sid
Carnatic	C	53831
Reggae203	B	53832
Topology112	A	53650
History 105	B	53666

Primary Key

sid	name	login	age	gpa
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

BATASAN INTEGRITAS

Dalam SQL

```
CREATE TABLE enrolled(sid CHAR(20),  
                        cid CHAR(20),  
                        grade CHAR(10),  
                        PRIMARY KEY (sid,cid),  
                        FOREIGN KEY (sid) REFERENCES Students)
```


QUERY DATA RELASIONAL

- Query merupakan pertanyaan mengenai data sedangkan jawabannya terdiri dari relasi baru yang memuat hasil.
- Bahasa Query merupakan bahasa khusus yang digunakan untuk menuliskan query.
- SQL merupakan bahasa query komersial yang paling populer untuk SMDB relasional.
- Contoh :

```
SELECT *  
FROM Students S  
WHERE S.age<18
```

QUERY DATA RELASIONAL

sid	name	login	age	gpa
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

QUERY DATA RELASIONAL

```
SELECT S.name, S.login  
FROM Students S  
WHERE S.age<18
```

name	login
Madayan	madayan@music
Guldu	guldu@music

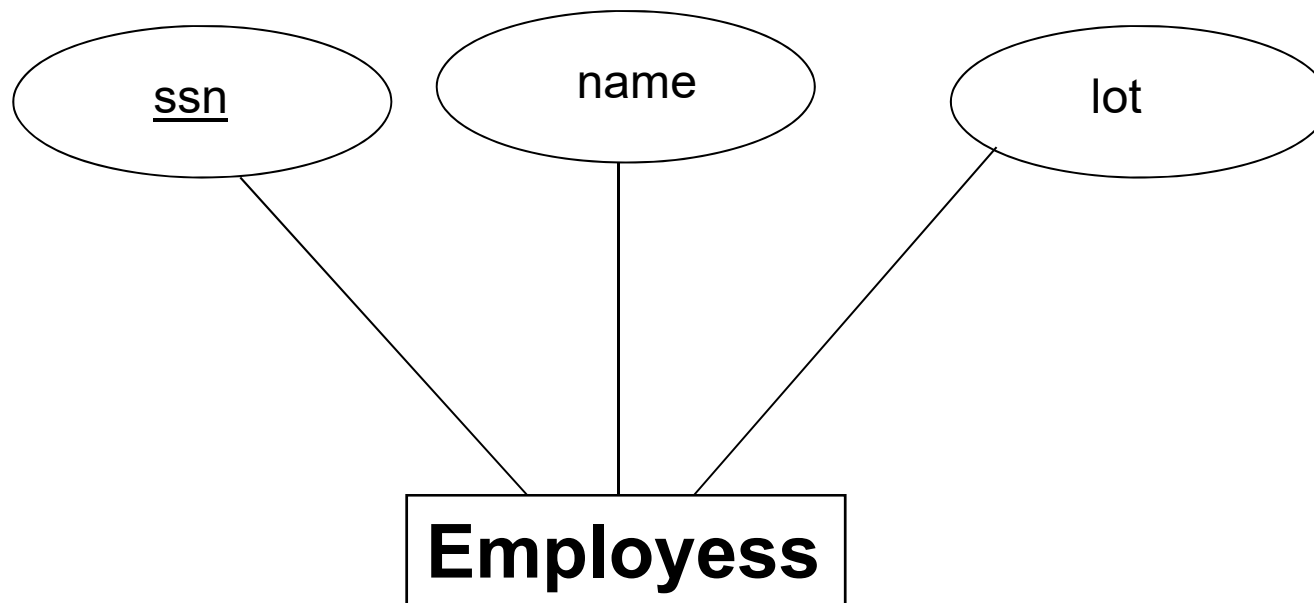
QUERY DATA RELASIONAL

```
SELECT S.name, E.cid  
FROM Students S, Enrolled E  
WHERE S.sid=E.sid AND E.grade='A'
```

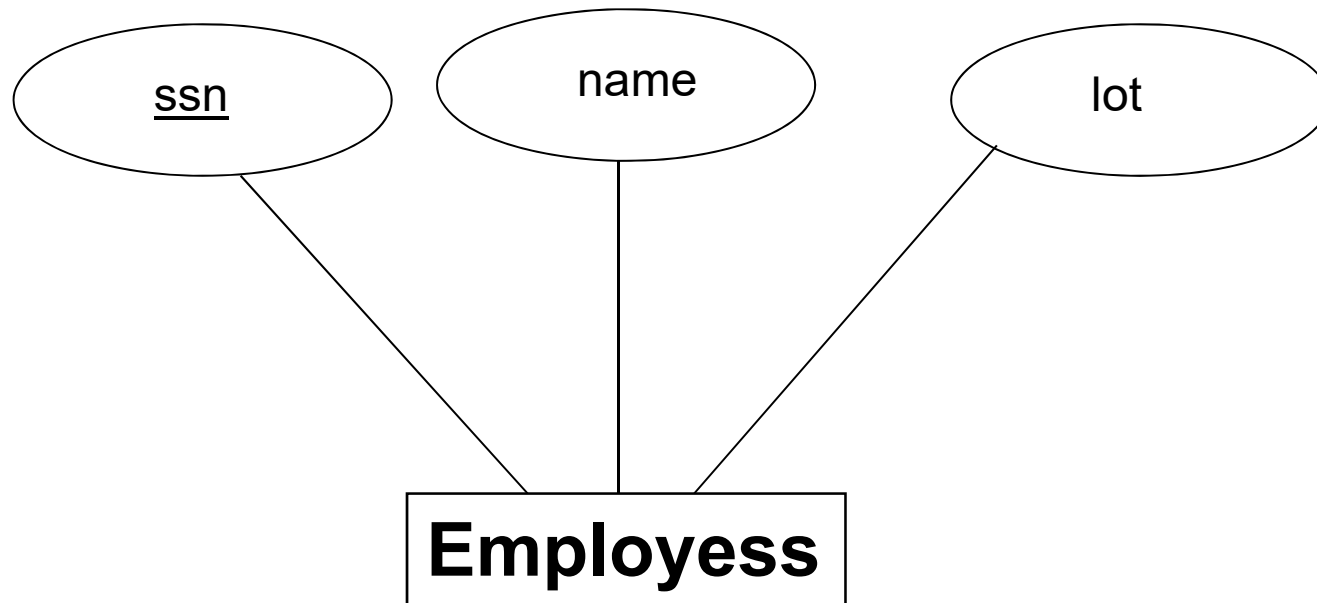
name	cid
Smith	Topology112

DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

- Model ER sangat sesuai untuk menggambarkan desain database awal yang memiliki level tinggi.
- Dilakukan pendekatan untuk menghasilkan skema database relasional yang sangat mendekati desain ER.



DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

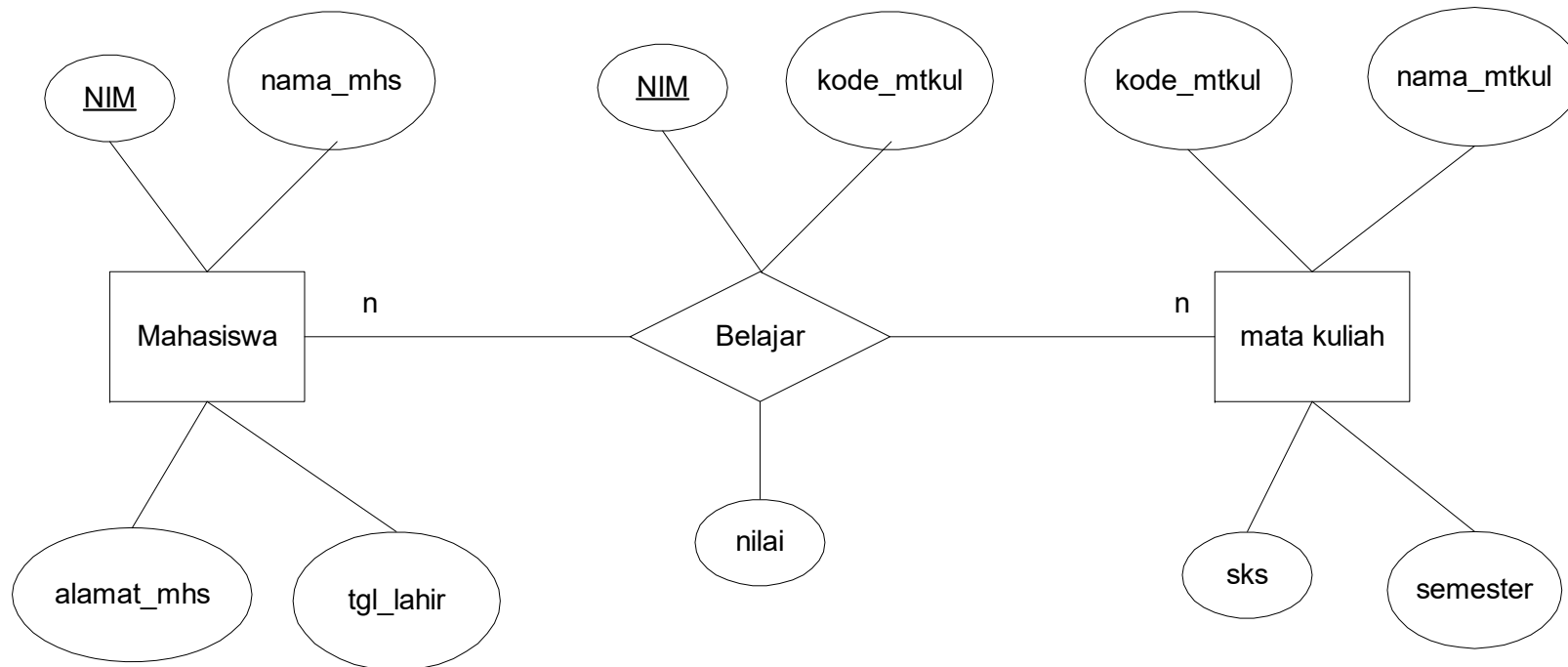


DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

ssn	name	Lot
123-22-3666	Attishoo	48
231-31-5368	Smiley	22
131-24-3650	Smethurst	35

```
CREATE TABLE Employees (ssn CHAR(11),  
                           name CHAR(30),  
                           lot INTEGER,  
                           PRIMARY KEY (ssn))
```

DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL



DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

```
CREATE TABLE mahasiswa (nim CHAR(9),  
                           nama_mhs CHAR(20),  
                           alamat_mhs CHAR(40),  
                           tgl_lahir    DATE,  
                           PRIMARY KEY (nim))
```

DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

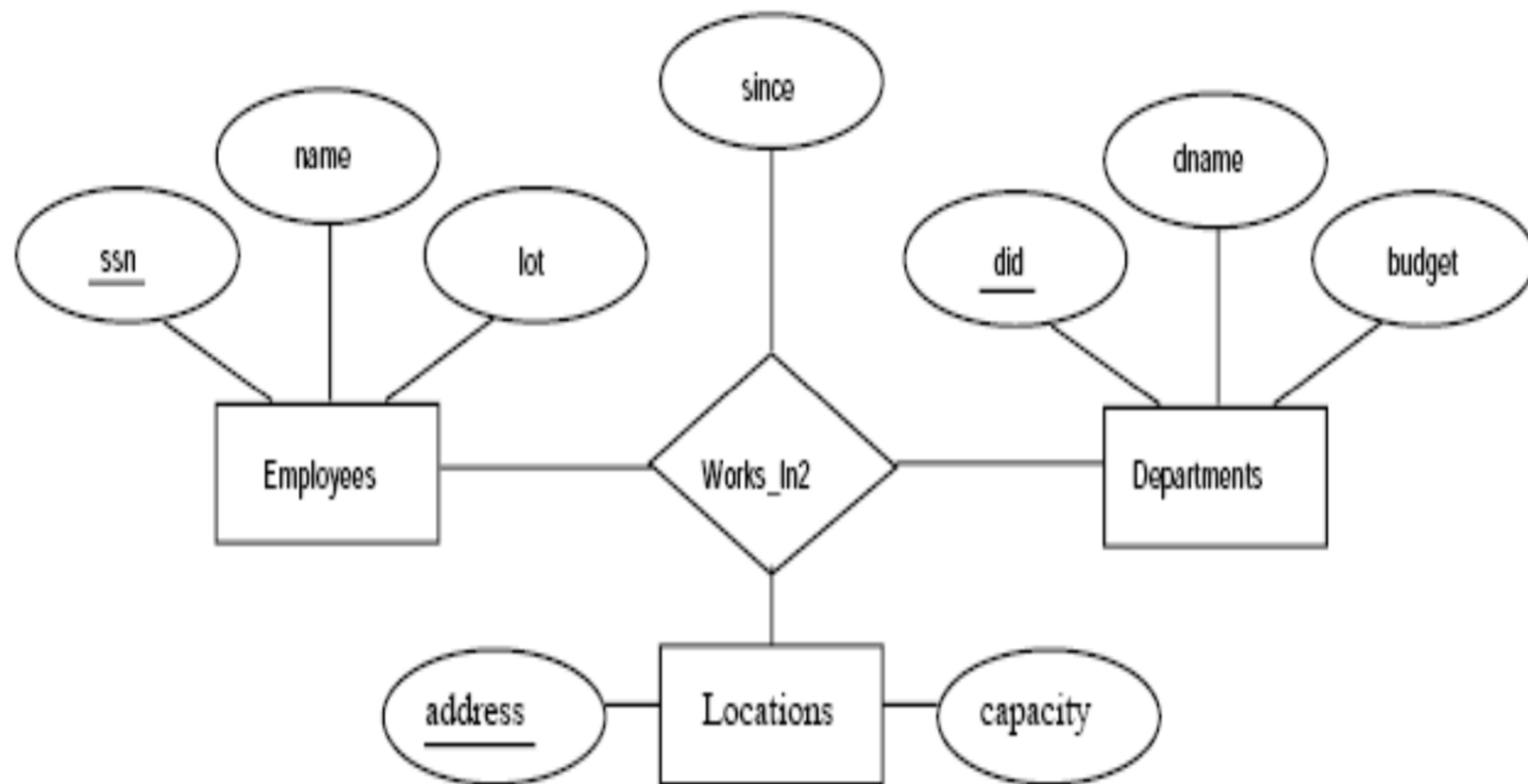
```
CREATE TABLE matakuliah (kode_mtkul CHAR(5),  
                           nama_mtkul CHAR(20),  
                           sks          INTEGER,  
                           semester     INTEGER ,  
                           PRIMARY KEY (kode_mtkul))
```

DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

```
CREATE TABLE belajar (nim      CHAR(9),  
                        kode_mtkul CHAR(5),  
                        nilai      CHAR(2),  
PRIMARY KEY (nim,kode_mtkul),  
FOREIGN KEY (nim) REFERENCES mahasiswa,  
FOREIGN KEY (kode_mtkul) REFERENCES  
matakuliah)
```

DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

Contoh:



DESAIN DATABASE LOGIS DARI E-R KE RELASIONAL

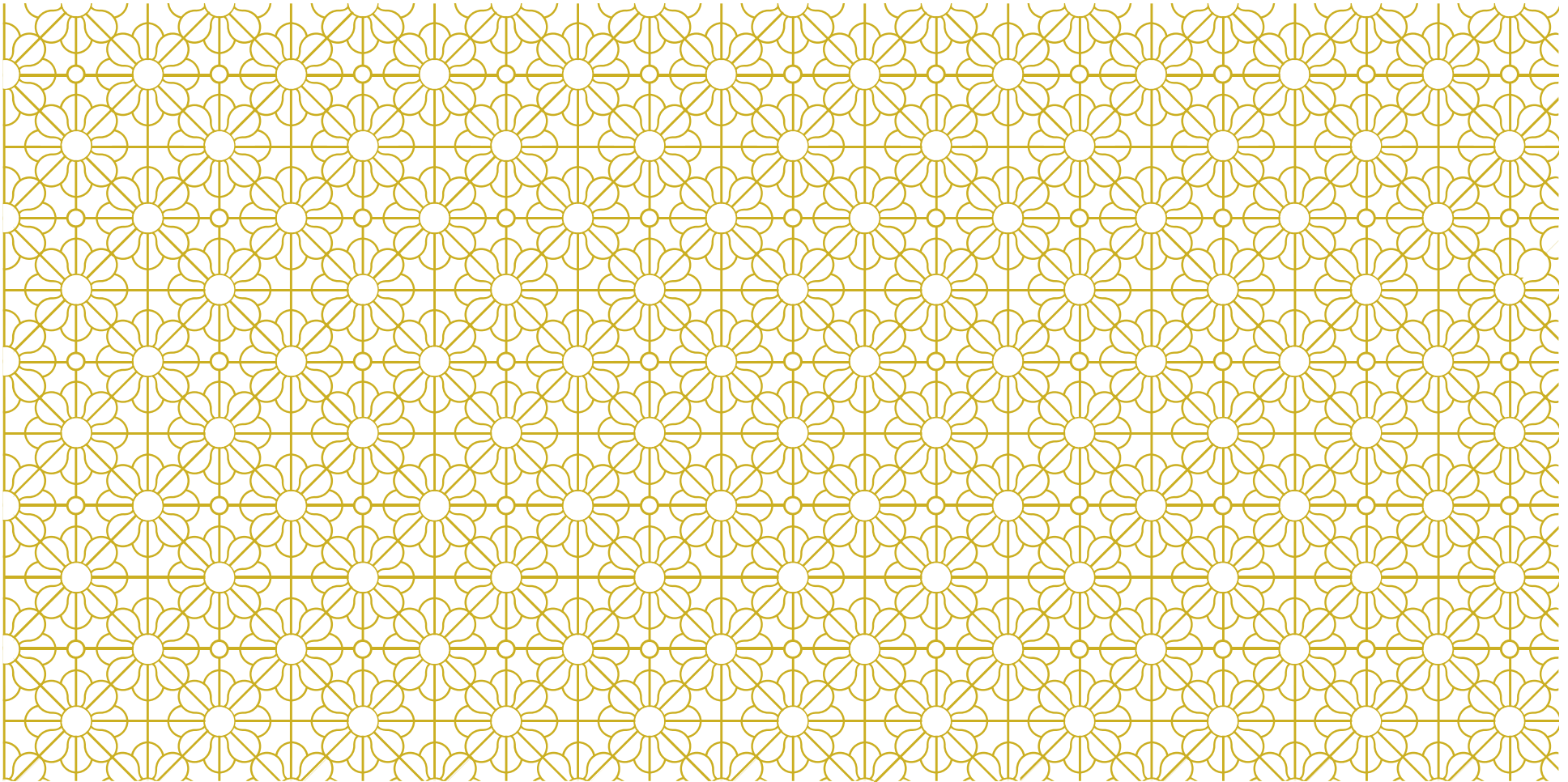
```
CREATE TABLE Works_In2 ( ssn      CHAR(11),  
                          did      INTEGER,  
                          address  CHAR(20),  
                          since    DATE,  
                          PRIMARY KEY (ssn, did, address),  
                          FOREIGN KEY (ssn) REFERENCES Employees,  
                          FOREIGN KEY (address) REFERENCES Locations,  
                          FOREIGN KEY (did) REFERENCES Departments )
```

STUDI KASUS

Sebuah perusahaan XYZ berencana untuk membuat sebuah database karyawannya.

Yang diharapkan dari database tersebut adalah bahwa perusahaan akan dengan cepat dapat memperoleh informasi tentang :

- Karyawan per-departemen.
- Karyawan yang lembur
- Gaji karyawan



BASIS DATA

ER-MODEL

ENTITY RELATIONSHIP DIAGRAM

- Sebuah teknik pemodelan data yang merepresentasikan gambar entitas dan relasi-relasi antar entitas di dalam sebuah sistem informasi
- Dibentuk oleh dua komponen utama:
 - Entitas (Entity)
 - Relasi (Relationship)yang dideskripsikan lebih detail dengan sejumlah atribut (properti)

ENTITAS (ENTITY)

Entity : merupakan obyek yang mewakili sesuatu dalam dunia nyata, baik secara fisik maupun secara konsep

contoh:

fisik : mobil, rumah, manusia, pegawai dsb

konsep : department, pekerjaan, mata kuliah dsb

Setiap entitas **pasti** memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut

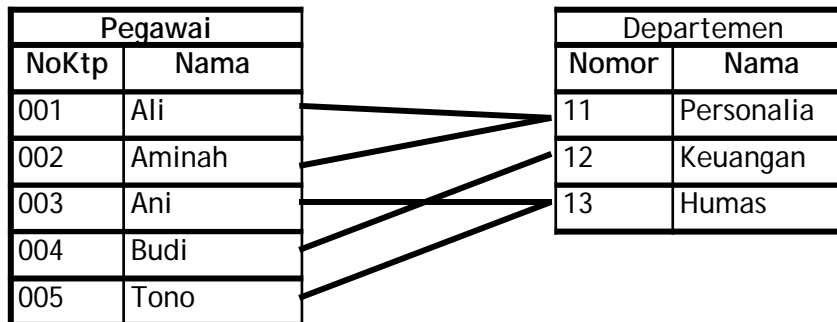
Notasi entitas:

NamaEntitas

RELASI (RELATIONSHIP)

Relasi menyatakan hubungan antar entitas, termasuk terhadap entitas itu sendiri (rekursif)

Contoh : entitas seorang pegawai dengan NoKTP: "001" dengan nama "Ali", memiliki relasi dengan sebuah data di entitas departemen dengan nomor=11 nama="Personalia"



KEY ATTRIBUTE

Adalah sebuah atau sekumpulan atribut yang membedakan data antara satu dengan lainnya (unik) dari seluruh data yang terdapat di dalam sebuah tabel.

Key Attribute dibagi menjadi 3:

- a. Superkey: Merupakan satu atau gabungan atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik
- b. Candidate Key: Adalah SuperKey yang jumlah atributnya minimal (paling sedikit).
- c. Primary Key: Adalah candidate key yang dipilih berdasarkan:
 - Seringnya dijadikan acuan
 - Lebih ringkas
 - Lebih menjamin keunikan key

SIMPLE ATTRIBUTE

Simple Attribute adalah attribut terkecil yang tidak bisa dipilah lagi.

Contoh:

Pada entitas Pegawai : Nama, Gaji

Pada entitas Proyek : Nomor, Nama, Lokasi

COMPOSITE ATTRIBUTE

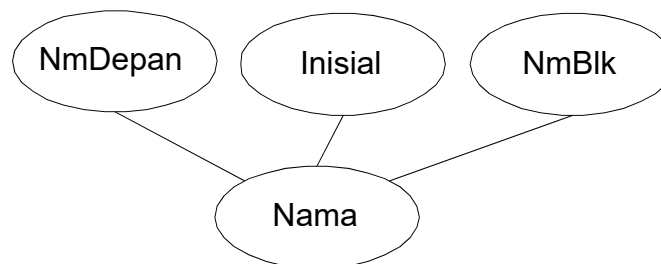
Composite Attribute adalah atribut yang dipilah-pilah lagi menjadi sub atribut yang masing-masing memiliki makna

Contoh:

Dalam sebuah kasus yang lain, bisa jadi merupakan Composite Attribute karena perlu dipilah-pilah lagi menjadi:

NmDepan, Inisial, NmBlk

Notasi:



SINGLE VALUED ATTRIBUTE

Single Valued Attribute merupakan attribut-attribut yang hanya memiliki sebuah nilai untuk sebuah data tunggal

Contoh:

Relasi “mengepalai” antara entitas Pegawai dengan entitas Departemen

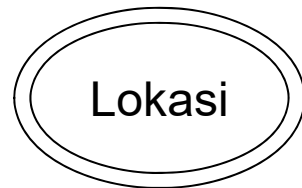
MULTI VALUED ATTRIBUTE

Multi Valued Attribute merupakan atribut-attribut yang bisa memiliki lebih dari satu nilai yang jenisnya sama dari sebuah data tunggal

Contoh:

Multi Valued: atribut lokasi pada Departemen

Notasi:



MANDATORY ATTRIBUTE

Mandatory attribute adalah attribut yang harus berisi data yang ada nilainya (tidak boleh kosong / NOT NULL)

Contoh:

Mandatory Attribute untuk entitas Pegawai adalah NoKTP atau Nama. Selain itu boleh kosong.

DERIVED ATTRIBUTE (ATTRIBUT TURUNAN)

Derived Attribute adalah atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan

Contoh:

atribut JmlPegawai pada entitas Departemen

Notasi:

JmlPegawai

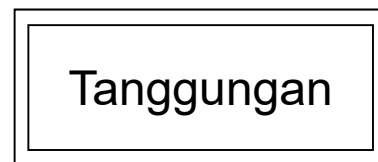
ENTITAS LEMAH (WEAK ENTITY)

Entitas lemah adalah entitas-entitas yang keberadaannya tergantung dari relasi terhadap entitas lain

Contoh:

entitas Tanggungan. Jika misalnya data seorang pegawai tidak ada atau ditiadakan, maka data tanggungan untuk pegawai tersebut tak akan ada

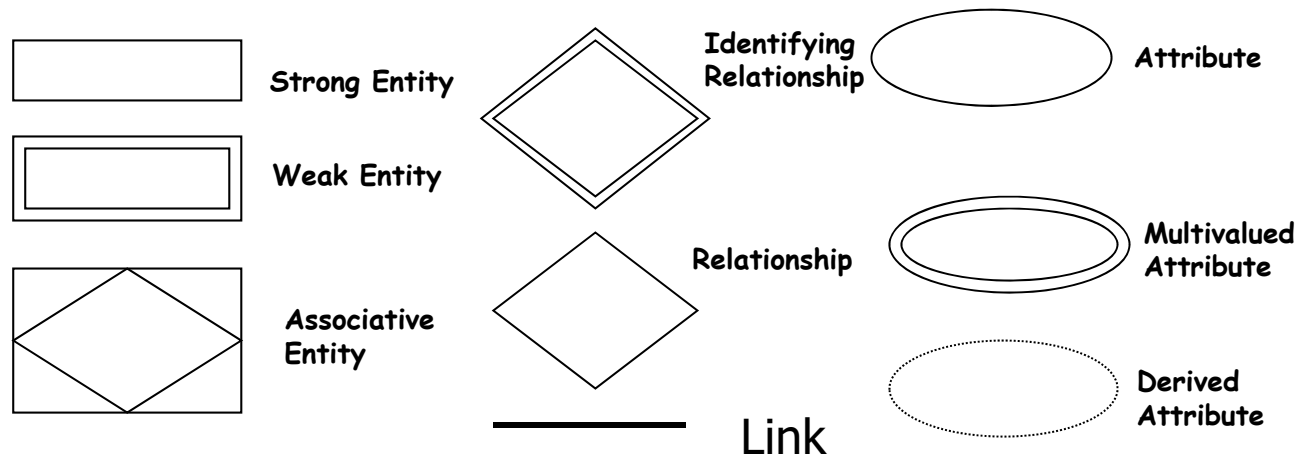
Notasi:



E-R Model (Model Keterhubungan Entitas)

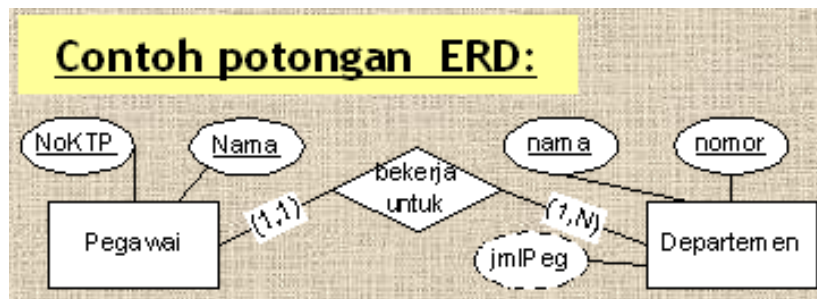
Semesta data di dunia nyata ditransformasikan ke dalam sebuah diagram dengan memanfaatkan perangkat konseptual disebut dengan ERD (Entity Relationship Diagram).

Simbol / Notasi E-R Diagram :



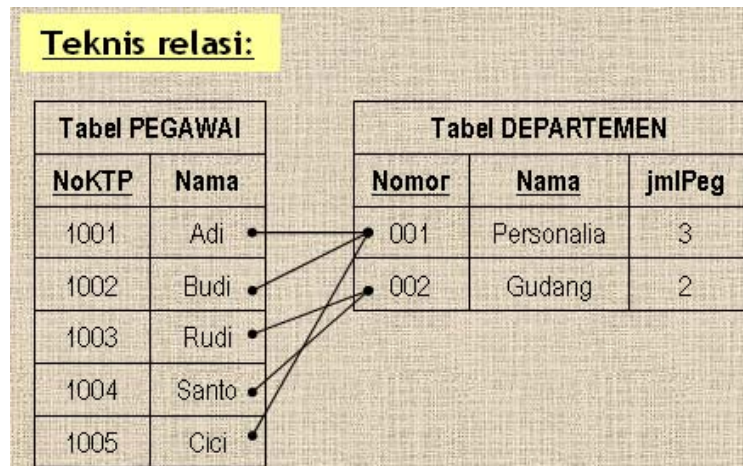
KONSEP FOREIGN KEY

Foreign key adalah kolom yang diambil dari primary key entitas lain yang menggunakan hubungan antar 2 tabel tsb.



Fakta

Adi, Budi dan Cici bekerja pada bagian Persinalia. Sedangkan Rudi dan Santo bekerja pada bagian Gudang



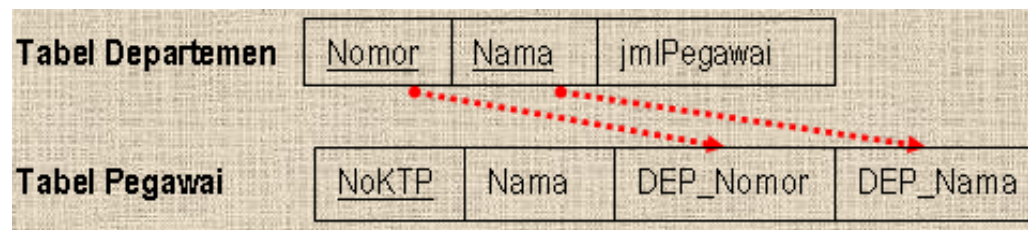
Dari fakta diatas dan panah relasi sebelah kiri, bagaimana cara menuliskan data yang menunjukkan hubungan/relasi antar tabel pegawai dan departemen?

KONSEP FOREIGN KEY

Ada 3 alternatif untuk menyimpan data relasi

1. Membuat tabel baru yang field-fieldnya berisi primary key kedua tabel
2. Primary key di tabel pegawai ditempatkan pula pada tabel departemen
3. Primary key dari tabel departemen ditempatkan pula pada tabel pegawai

Alternatif 1 dan 2 tidak dipakai karena mengakibatkan pengulangan yang tidak perlu (boros). Alternatif ke-3 dipilih, sehingga struktur tabelnya menjadi:



Kolom DEP_Nomor dan DEP_Nama pada tabel Pegawai disebut dengan foreign key, karena diambil dari primary key entitas lain

DERAJAT KARDINALITAS RELASI (CARDINALITY RATIO)

Kardinalitas relasi menunjukkan jumlah maksimum data entitas yang dapat berelasi dengan entitas lain.

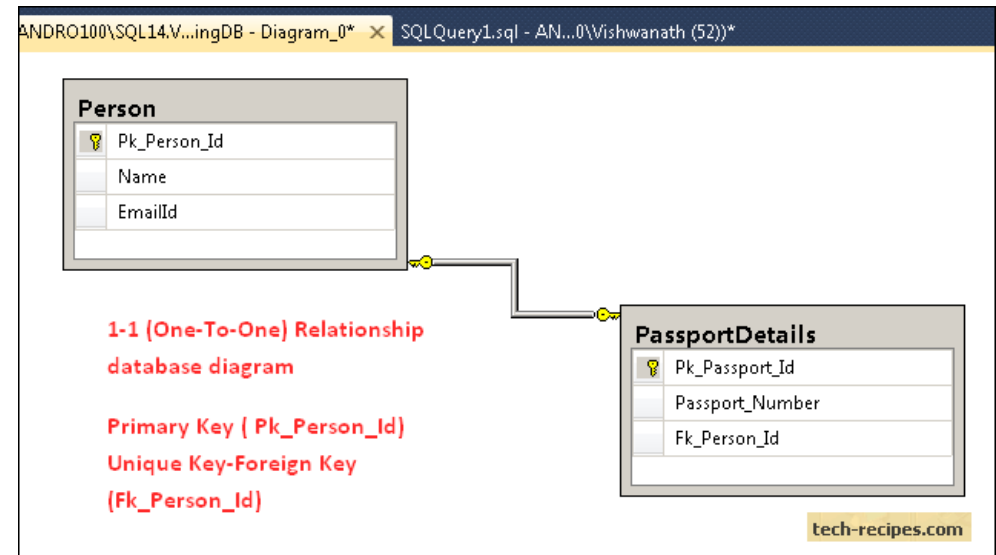
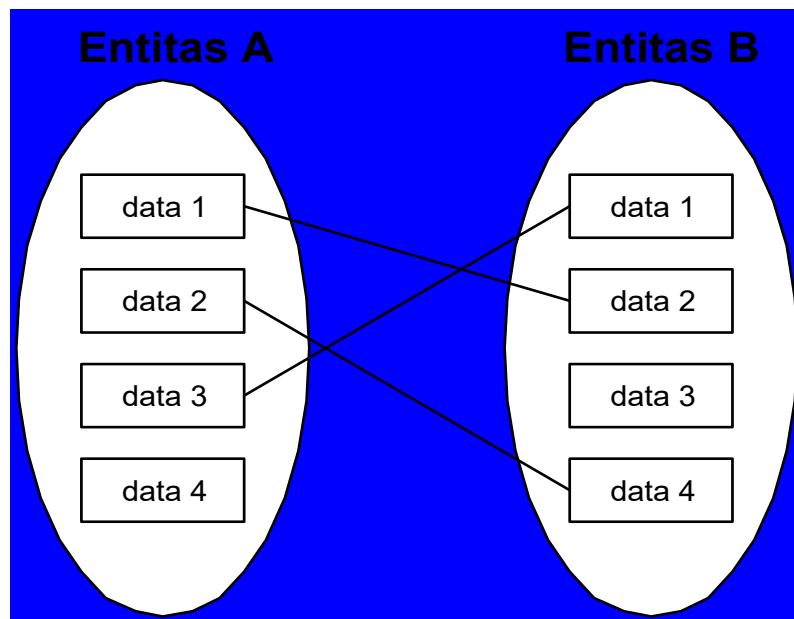
Contoh:

[Table Relationships in SQL Server](#)

SATU KE SATU (ONE TO ONE)

Setiap data pada entitas A berhubungan dengan maksimal satu data pada entitas B, begitu pula sebaliknya

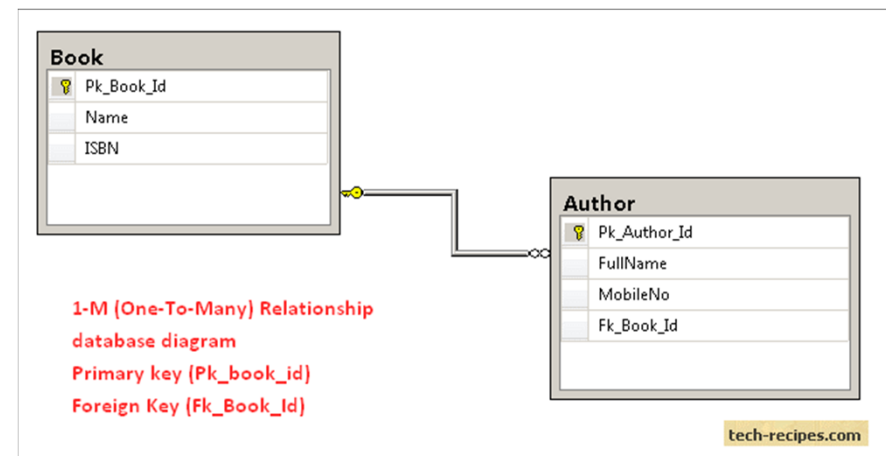
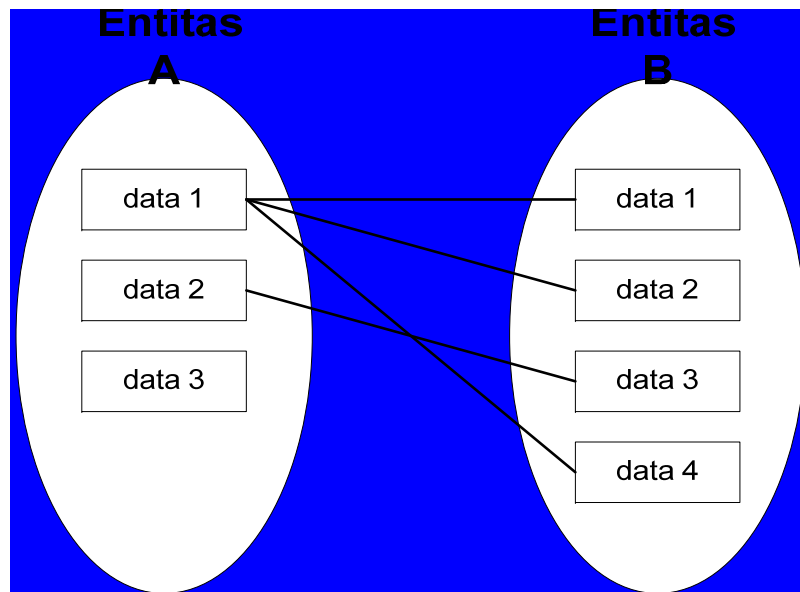
Contoh: relasi “mengepalai” antara entitas Pegawai dengan entitas Departemen



SATU KE BANYAK (ONE TO MANY)

Setiap data pada entitas A bisa berhubungan dengan banyak data pada entitas B, tetapi data pada entitas B berhubungan maksimal hanya dengan sebuah data di A

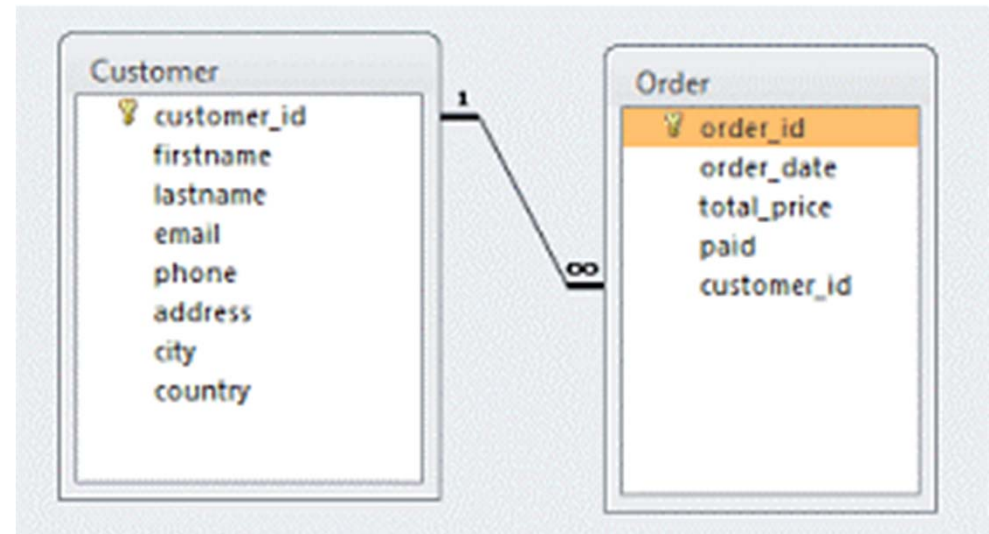
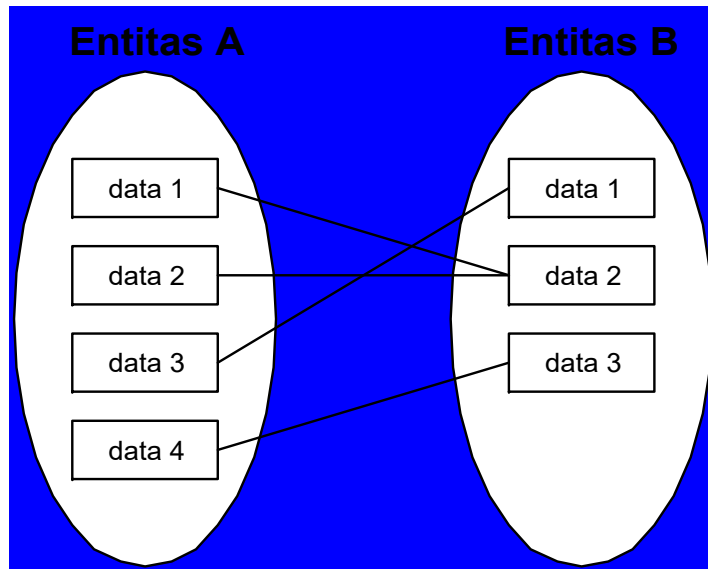
Contoh: relasi “menanggung” antara entitas pegawai terhadap entitas tanggungan



BANYAK KE SATU (MANY TO ONE)

Merupakan kebalikan dari relasi *satu-ke-banyak*

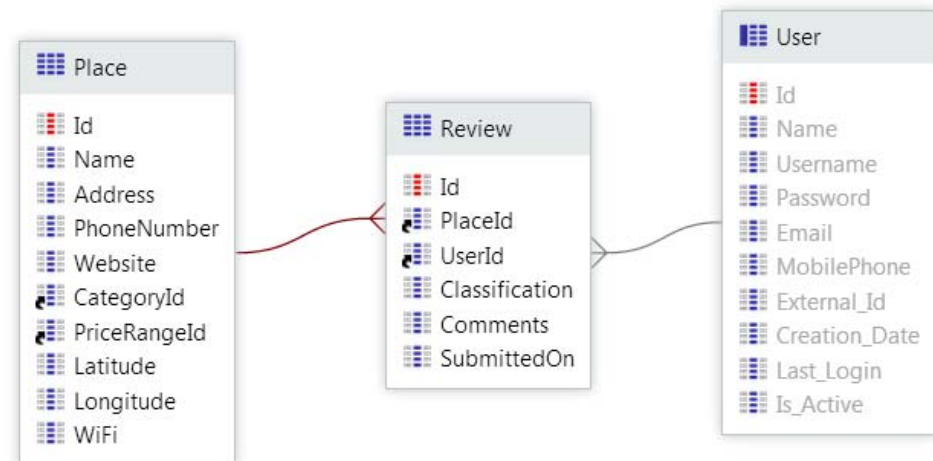
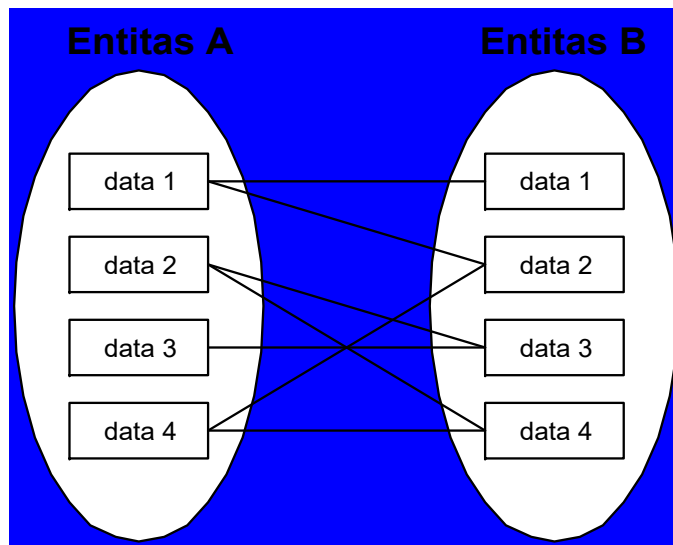
Contoh: relasi “bekerja untuk” pada entitas pegawai terhadap entitas departemen



BANYAK KE BANYAK (MANY TO MANY)

Setiap data pada entitas A bisa berhubungan dengan banyak data pada entitas B, demikian pula sebaliknya

Contoh: relasi “bekerja pada” pada entitas Pegawai terhadap entitas proyek



BATASAN PARTISIPASI (PARTICIPATION CONSTRAINT)

Menentukan apakah keberadaan sebuah entitas tergantung pada hubungannya ke entitas lain melalui jenis relasinya

Ada 2 jenis:

1. **Partisipasi Total** → partisipasi pegawai dalam relasi “bekerja untuk”, memberikan arti bahwa setiap data pada pegawai, harus direlasikan pada sebuah departemen. Atau tidak ada data pegawai yang tidak berelasi dengan data departemen
2. **Partisipasi sebagian** → partisipasi pegawai dalam relasi “mengepalai” terhadap departemen, tidak setiap data pada pegawai dihubungkan dengan departemen dengan relasi ini. Atau tidak semua pegawai mengepalai departemen

DERAJAT RELASI MINIMUM

- ❖ Menunjukkan hubungan (korespondensi) minimum yang boleh terjadi dalam sebuah relasi antar entitas.
- ❖ Notasi (x,y) pada relasi menunjukkan derajat minimum (x) dan derajat maksimum (y) pada sebuah relasi
- ❖ Partisipasi total bisa dinotasikan dengan memberikan derajat relasi minimum $(x) = 1$

Contoh:

pada relasi “bekerja untuk” pada entitas pegawai terhadap departemen Derajat relasi minimumnya adalah satu

NOTASI LAIN UNTUK RELASI









<u>Notasi</u>	<u>Derajat relasi</u> <u>minimum-maksimum</u>
 atau 	(0, n)
 atau 	(1, n)
 atau 	(0,1)
 atau 	(1,1)

FIGURE 2.3

The Chen and Crow's Foot notations

Chen Notation

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



Crow's Foot Notation

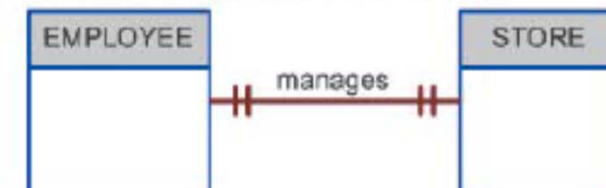
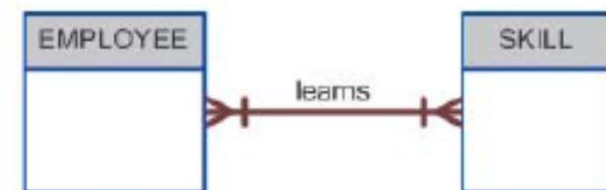
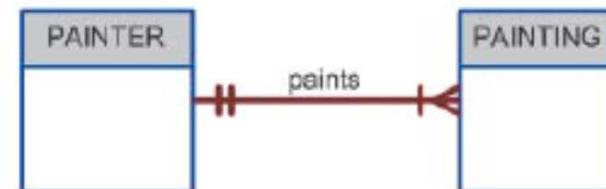
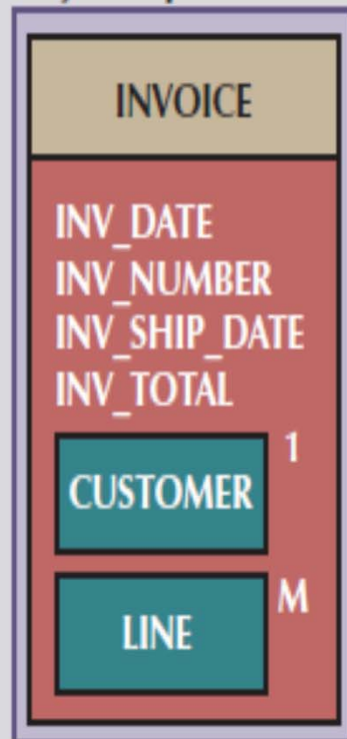


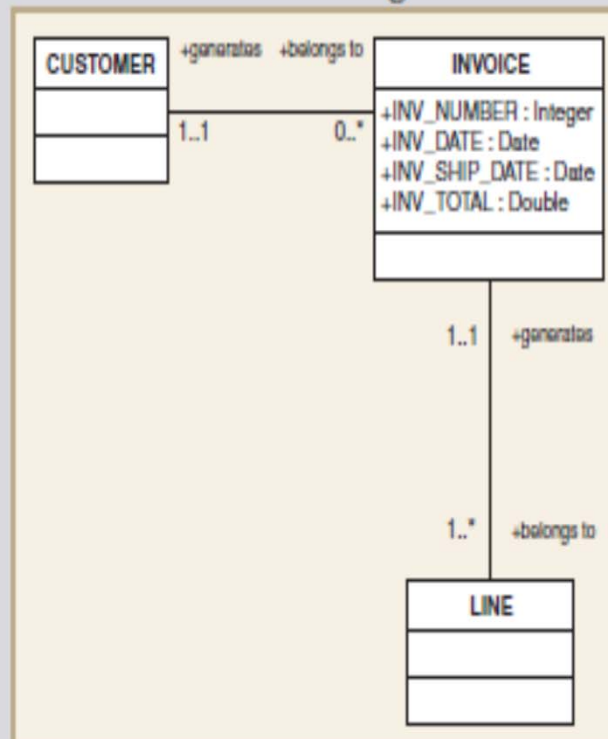
FIGURE
2.4

A comparison of OO, UML, and ER models

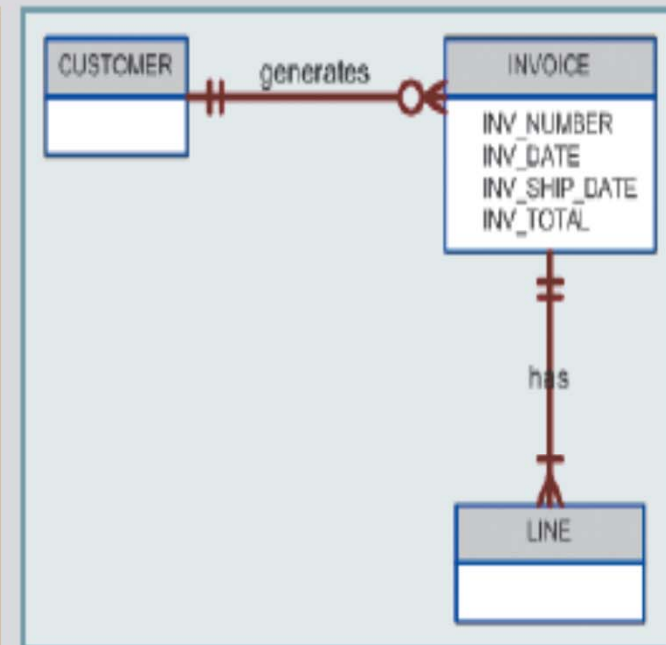
Object Representation



UML Class Diagram



ER Model



MACAM RELASI

- relasi Biner (binary relation)
- relasi tunggal (unary relation)
- relasi Multi entitas (n-ary relation)
- relasi ganda (redundant relation)

RELASI BINER (BINARY RELATION)

Merupakan relasi yang terbentuk antar 2 buah entitas

Contoh:

relasi “bekerja pada” pada entitas pegawai terhadap entitas proyek

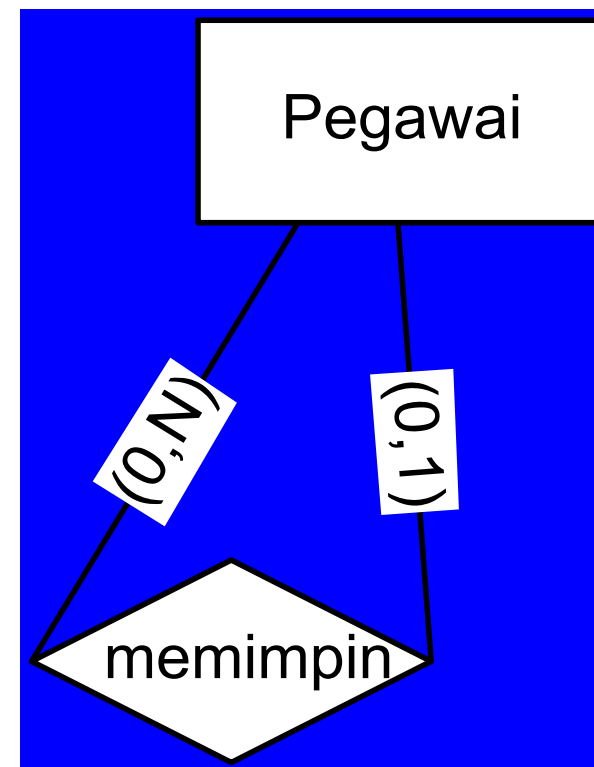


RELASI TUNGGA (UNARY RELATION)

Merupakan relasi yang terjadi dari sebuah entitas ke entitas yang sama

Contoh:

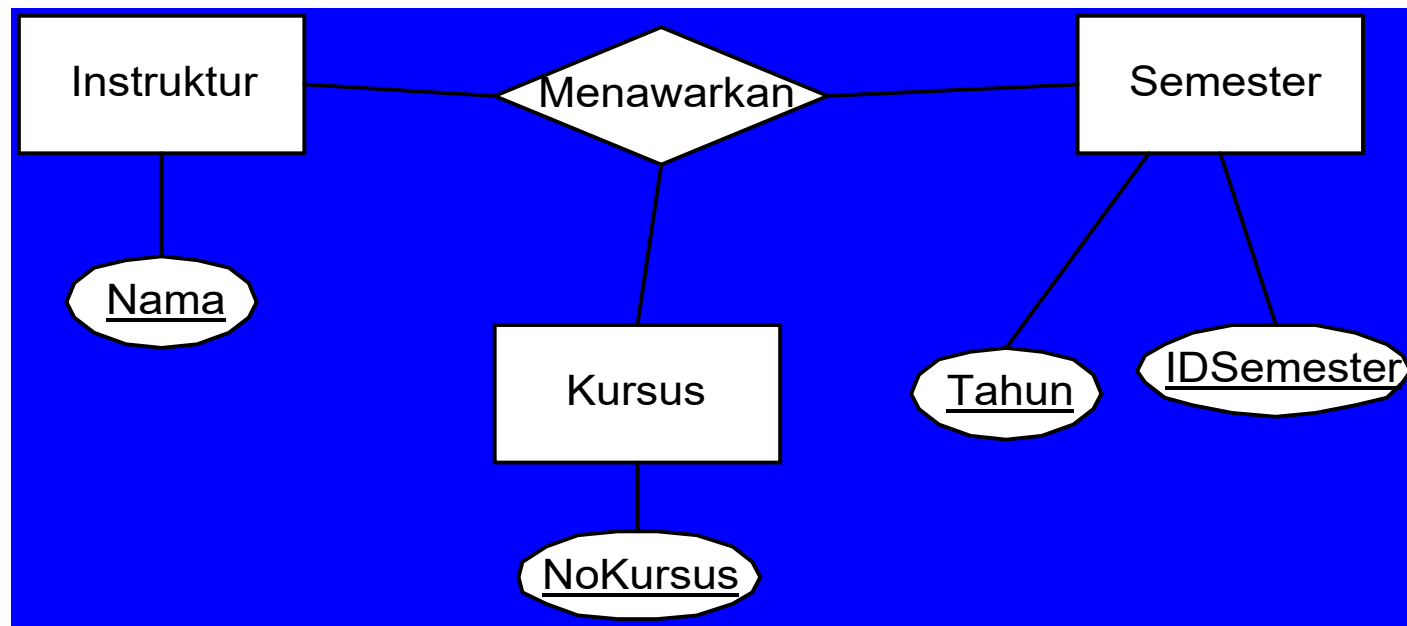
relasi memimpin pada entitas pegawai



RELASI MULTI ENTITAS (N-ARY RELATION)

Merupakan relasi dari 3 buah entitas atau lebih. Seharusnya dihindari, karena akan mengaburkan derajat relasi yang ada dalam relasi.

Contoh:



RELASI GANDA (REDUNDANT RELATION)

Relasi yang jumlahnya lebih dari satu untuk dua buah entitas

Contoh:

