



BASIS DATA

# PEMETAAN ERD KE MODEL RELASIONAL

*TIM AJAR BASIS DATA JTI-POLINEMA*

# POKOK BAHASAN



- Pengertian
- Sejarah
- Keuntungan
- Contoh Model Relasional
- Relational Key
- Relational Integrity
- Mapping ERD ke Model Relasional



# PENDAHULUAN

- Komputer mempunyai fungsi utama untuk menyimpan dan mengelola informasi
- Perlu dilakukan tata cara mengatur informasi tersebut
- Cara yang sederhana dan memudahkan data untuk diakses dan dikelola
- Model relasional digagas untuk mengorganisasi ke data ke dalam banyak tabel dua dimensi yang saling berelasi

# SEJARAH

Model relasional pertama kali dikenalkan oleh Codd, pada tahun 1971

Sejak itu model relasi memainkan peranan yang sangat penting dalam berbagai perancangan basis data. Ada tiga alasan mengapa model relasi mempunyai peranan penting dalam perancangan basis data yaitu :

Relasi merepresentasikan struktur data yang dapat dimengerti oleh user maupun designer

Model relasional mendefinisikan salah satu kriteria perancangan basis data yang penting yaitu relasi bentuk normal

Struktur data yang direpresentasikan oleh relasi dapat segera dikonversikan & diimplementasikan ke RDBMS

Pada tahun 1985, Codd menerbitkan daftar 12 peraturan untuk mendefinisikan sistem basis data relasional, karena kekhawatiran bahwa banyak vendor memasarkan produk sebagai "relasional" walaupun produk tersebut tidak memenuhi standar relasional minimum

# CODD'S RELATIONAL DATABASE

TABLE  
3.8

Dr. Codd's 12 Relational Database Rules

RULE	RULE NAME	DESCRIPTION
1	<b>Information</b>	All information in a relational database must be logically represented as column values in rows within tables.
2	<b>Guaranteed Access</b>	Every value in a table is guaranteed to be accessible through a combination of table name, primary key value, and column name.
3	<b>Systematic Treatment of Nulls</b>	Nulls must be represented and treated in a systematic way, independent of data type.
4	<b>Dynamic On-Line Catalog Based on the Relational Model</b>	The metadata must be stored and managed as ordinary data, that is, in tables within the database. Such data must be available to authorized users using the standard database relational language.
5	<b>Comprehensive Data Sublanguage</b>	The relational database may support many languages. However it must support one well-defined declarative language with support for data definition, view definition, data manipulation (interactive and by program), integrity constraints, authorization, and transaction management (begin, commit, and rollback).
6	<b>View Updating</b>	Any view that is theoretically updatable must be updatable through the system.
7	<b>High-Level Insert, Update and Delete</b>	The database must support set-level inserts, updates, and deletes.
8	<b>Physical Data Independence</b>	Application programs and ad hoc facilities are logically unaffected when physical access methods or storage structures are changed.
9	<b>Logical Data Independence</b>	Application programs and ad hoc facilities are logically unaffected when changes are made to the table structures that preserve the original table values (changing order of column or inserting columns).
10	<b>Integrity Independence</b>	All relational integrity constraints must be definable in the relational language and stored in the system catalog, not at the application level.
11	<b>Distribution Independence</b>	The end users and application programs are unaware and unaffected by the data location (distributed vs. local databases).
12	<b>Nonsubversion</b>	If the system supports low-level access to the data, there must not be a way to bypass the integrity rules of the database.
	<b>Rule Zero</b>	All preceding rules are based on the notion that in order for a database to be considered relational, it must use its relational facilities exclusively to manage the database.

# KEUNTUNGAN



1

- Memiliki bentuk sederhana

2

- Mempermudah operasi data (query, update/edit, delete).



# TOOLS MODEL RELASIONAL

- DBMS
  - DB2 (IBM)
  - Rdb/VMS (Digital Equipment Corporation)
  - Oracle (Oracle Corporation)
  - Informix (Informix Corporation)
  - Ingres (ASK Group Inc)
  - Sybase (Sybase Inc)
- PC
  - R:Base Family
  - dBase Family
  - Microsoft SQL
  - Visual FoxPro

## ISTILAH-ISTILAH MODEL RELASIONAL

<i>Relation/table/file</i>	• Representasi tabel yang terdiri dari sejumlah baris dan kolom
<i>Attribute/column/field</i>	• kolom pada tabel
<i>Tuple/row/record</i>	• Baris pada tabel
<i>Domain</i>	• Himpunan nilai dari satu atau lebih <i>attribute</i>
<i>Degree</i>	• Banyaknya <i>attribute</i> /kolom pada tabel
<i>Cardinality</i>	• Banyaknya <i>tuple</i> /baris pada tabel
<i>Relational Basis Data</i>	• Kumpulan relasi ternormalisasi dengan nama relasi yang jelas





## CONTOH BASIS DATA MODEL RELASIONAL

**Tabel Customer**

customer-id	customer-name	customer street	customer city
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

**Tabel Account**

account-number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

**Tabel Depositor**

customer-id	account-number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201



## CONTOH BAGIAN MODEL RELASIONAL

Relation	Attribute	Degree	Tuple	Cardinality
Tabel Customer	customer-id customer-name customer-street customer-city	4	Baris 1 – Baris 7	7
Tabel Depositor	customer-id account-number	2	Baris 1 – Baris 8	8
Tabel Account	account-number balance	2	Baris 1 – Baris 7	7

# RELASI BASIS DATA



- Skema Relasi  
Nama relasi didefinisikan oleh himpunan pasangan atribut dan nama domain
- Skema Basis Data relational  
Himpunan skema relasi dengan nama yang berbeda

# SIFAT-SIFAT RELASI



- Nama relasi berbeda satu sama lain dalam skema relasional
- Setiap sel(baris,kolom) dari relasi berisi satu nilai atomik atau nilai tunggal
- Setiap atribut memiliki nama yang berbeda
- Nilai suatu atribut berasal dari domain yang sama
- Setiap *tuple* adalah berbeda, dan tidak ada duplikasi *tuple*

# Fase-fase desain basis data sederhana

## Phase 1 :

Requirements,  
Collection and  
Analysis

## Phase 2 :

Conceptual Database Design

## Phase 3 :

Choice of DBMS

## Phase 4 :

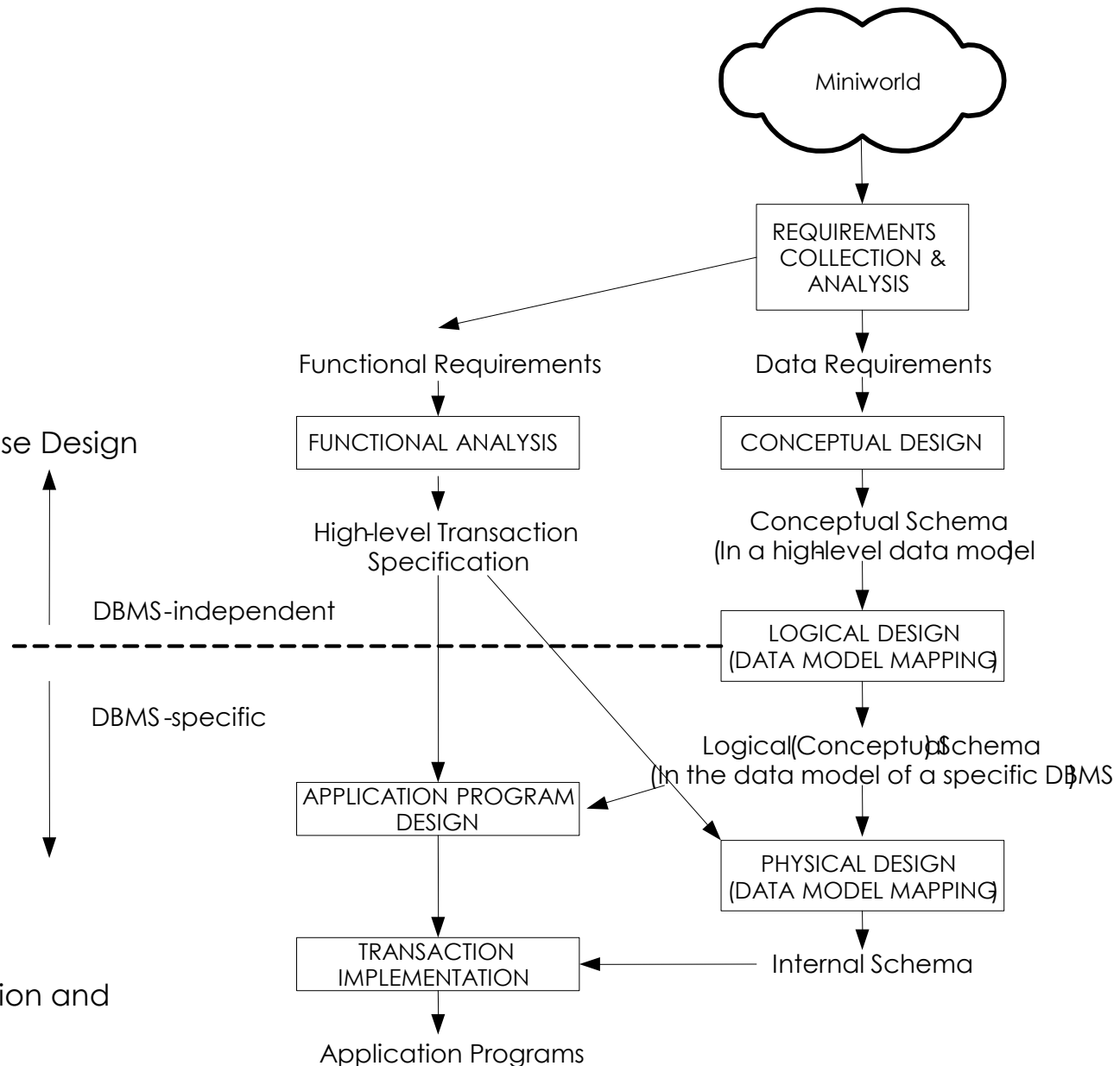
Data Model  
Mapping (Logical  
design)

## Phase 5 :

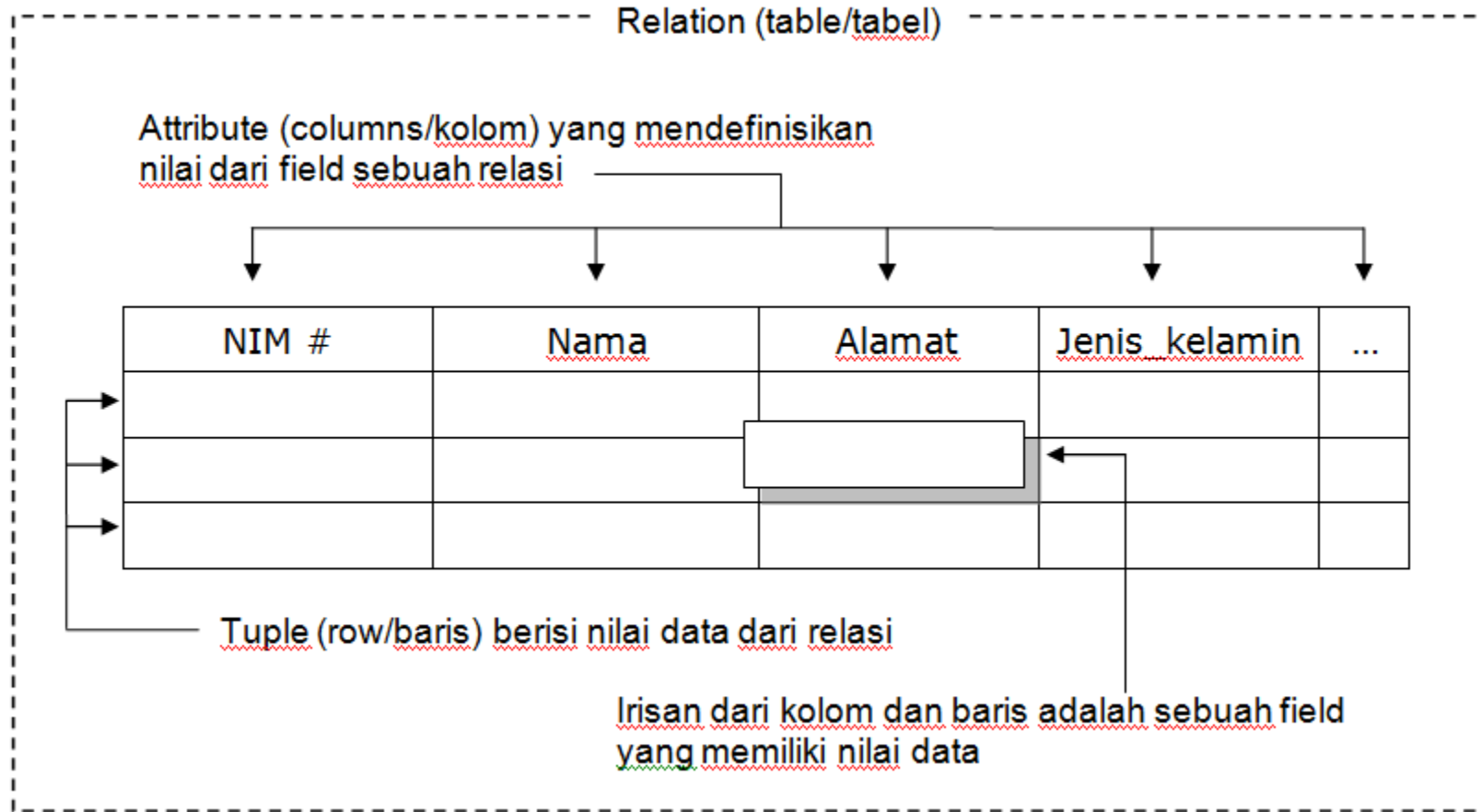
Physical Design

## Phase 6 :

System Implementation and  
Tuning



# RELASIONAL





# RELATIONAL KEY

## **Superkey**

- Sebuah atribut (atau kombinasi atribut) secara unik mengenali setiap entitas dalam sebuah tabel.

## **Candidate key**

- Sebuah *superkey* minimal, yaitu *superkey* yang tidak merupakan bagian atribut dari suatu *superkey*.

## **Primary key**

- *Candidate key* yang terpilih untuk mengenali secara unik seluruh nilai atribut pada sebuah baris. Tidak boleh kosong.

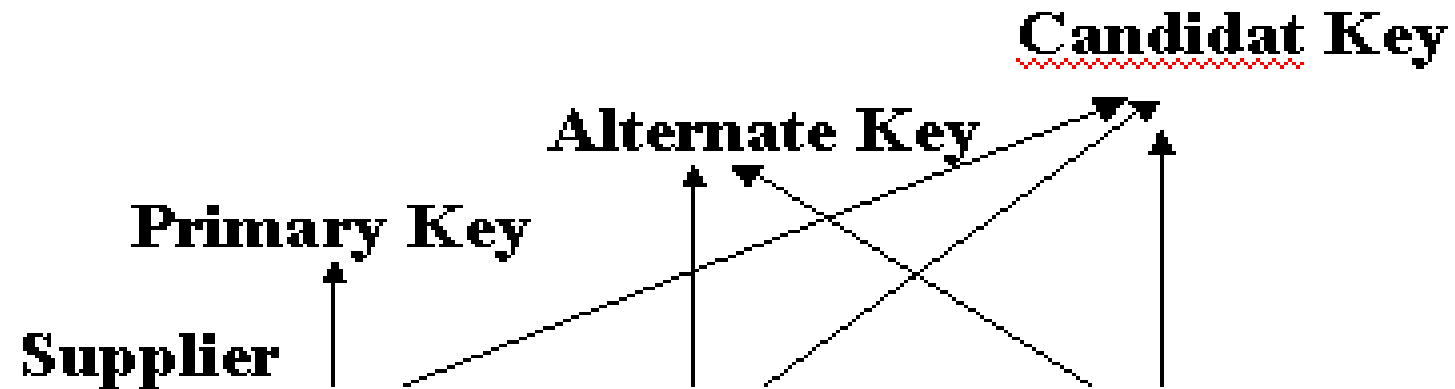
## **Secondary key**

- Sebuah atribut (atau kombinasi atribut) secara paksa digunakan untuk tujuan pengambilan data.

## **Foreign key**

- Sebuah atribut (atau kombinasi atribut) dalam sebuah tabel dimana nilainya cocok dengan *primary key* pada tabel lainnya.

# ILUSTRASI



KD_SUPP	NAMA_SUPP	ALAMAT
S0001	PANASONIC	DEPOK
S0002	SANYO	BOGOR
S0003	AKARI	BEKASI



# CONTOH RELATIONAL KEY



Relasi	Superkey	Candidate Key	Primary Key	Foreign Key	Alternatif key
Tabel customer	(customer-id) (customer-id,customer-name) (customer-id,customer-name,customer-city)	customer-id customer-name	customer-id	customer-city  (dengan syarat terdapat tabel city yang memiliki field city-id)	customer-name

# RELATIONAL INTEGRITY



- Null
  - Representasi nilai atribut yang tidak diketahui atau tidak digunakan dalam *tuple*
  - Berkaitan dengan ketidaklengkapan atau pengecualian data
  - Representasi tidak adanya suatu nilai, dan tidak sama dengan nol atau spasi
- *Entity Integrity*
  - Pada relasi dasar, tidak ada atribut ataupun primary key yang bernilai NULL
- *Referential Integrity*
  - Jika terdapat foreign key dalam suatu relasi, maka foreign key tersebut dibandingkan dengan candidate key dan beberapa tuple pada relasi itu sendiri
- *Enterprise Constraint*
  - Aturan tambahan yang diberikan oleh user atau DBA



# CONTOH RELATIONAL INTEGRITY

Null	Entity Integrity	Referential Integrity	Enterprise Constraint
Field customer-description (diasumsikan terdapat kolom customer-description pada tabel customer)	Field customerId pada tabel customer	Field customer-id pada tabel depositor	Jika jumlah account-number untuk setiap customer-id diberi batasan misalkan 5 account-number, maka lebih dari jumlah tersebut tidak diperbolehkan

# PERATURAN MAPPING (ERD KE MODEL RELASIONAL)

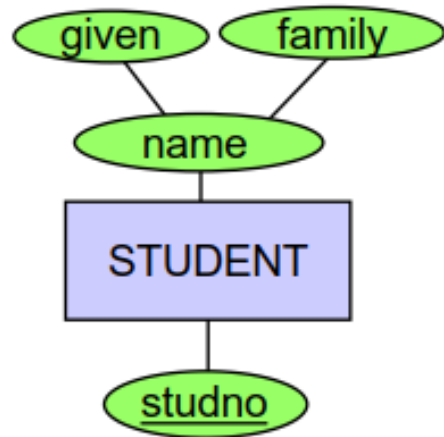


Untuk setiap tipe entitas yang membuat relasi

- Setiap atribut atom dari tipe entitas menjadi atribut relasi
- Atribut komposit mencakup semua atribut atom
- Atribut turunan tidak disertakan dalam field
- Contoh relasi adalah himpunan bagian dari cross product dari domain atribut
- Atribut dari kunci entitas membentuk kunci utama relasi

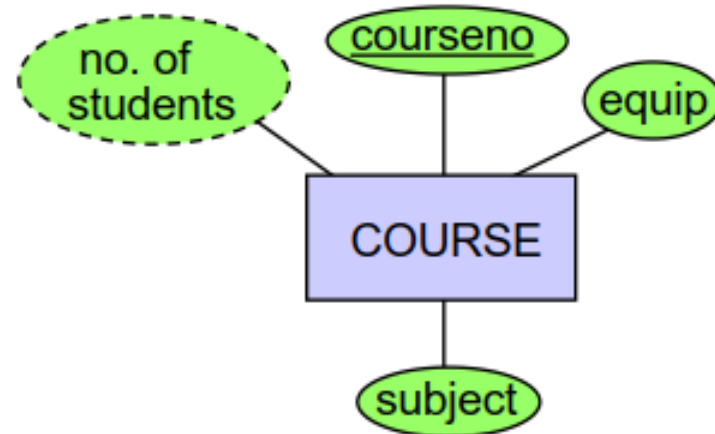
# TRANSFORMASI DASAR

Setiap himpunan entitas akan diimplementasikan sebagai sebuah tabel



Tabel Student

<u>studno</u>	givenname	familyname

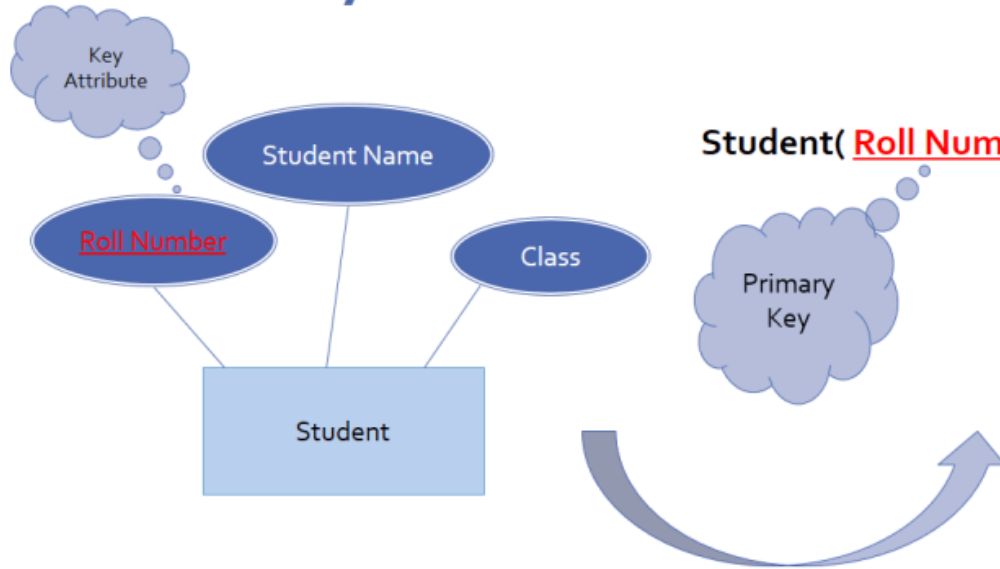


Tabel Course

<u>courseno</u>	subject	equip

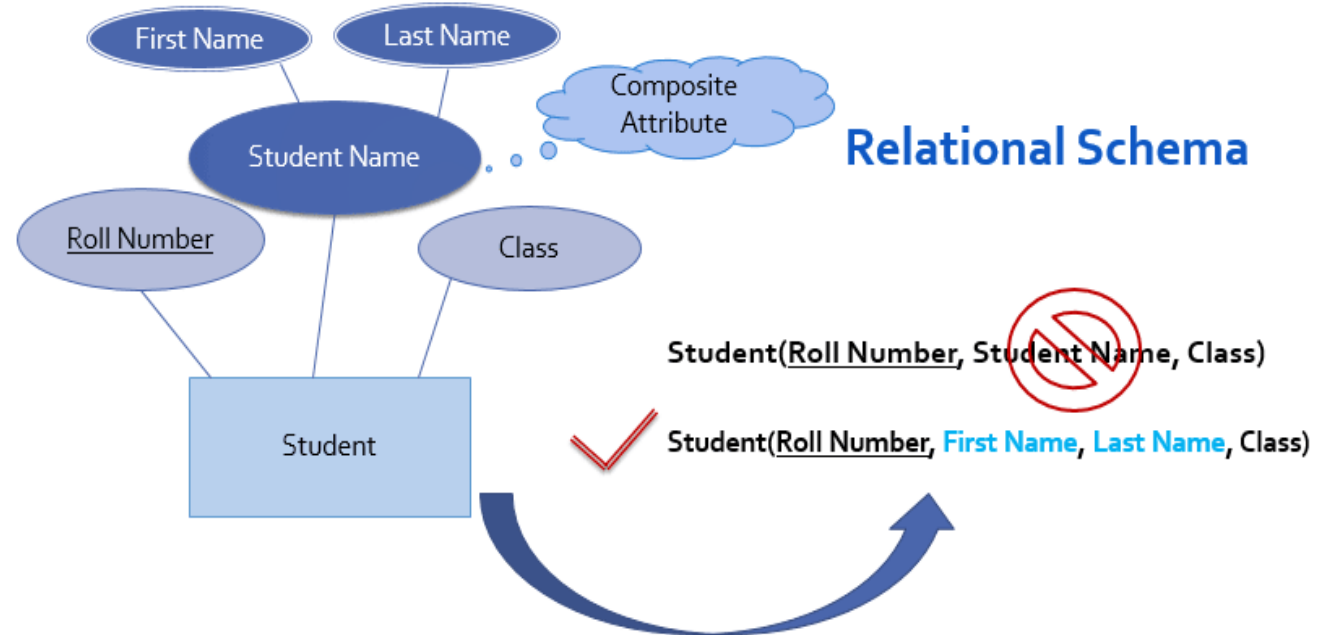
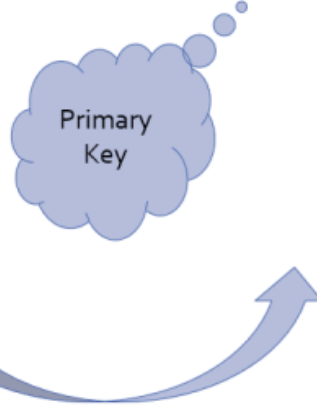
# TRANSFORMASI DASAR

## 1. Entity Set



## Relational Schema

Student(Roll Number, Student Name, Class)

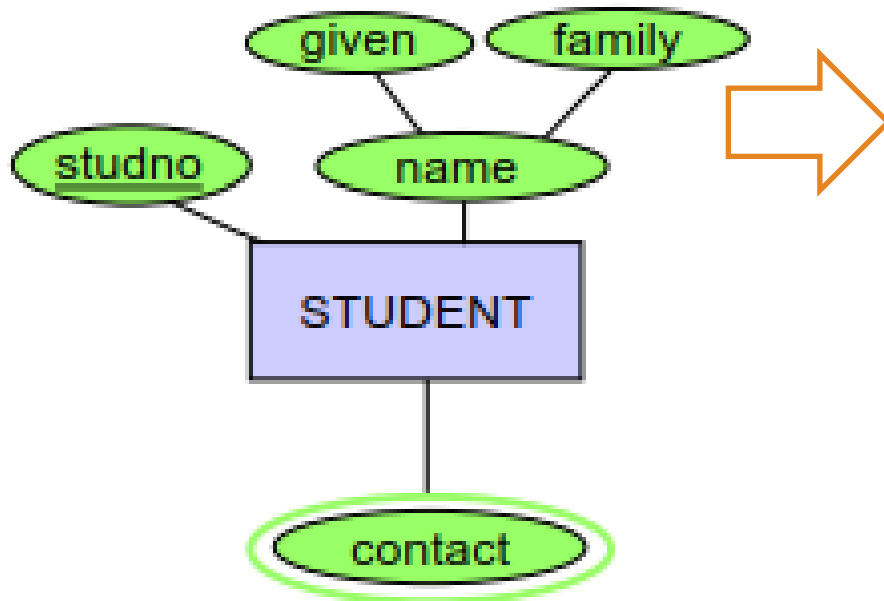


Student(Roll Number, ~~Student Name~~, Class)

✓ Student(Roll Number, First Name, Last Name, Class)

# ATRIBUT MULTIVALUE

Atribut bernilai banyak membuat sebuah tabel baru dengan mengambil kunci utama sebuah entitas



Tabel Student

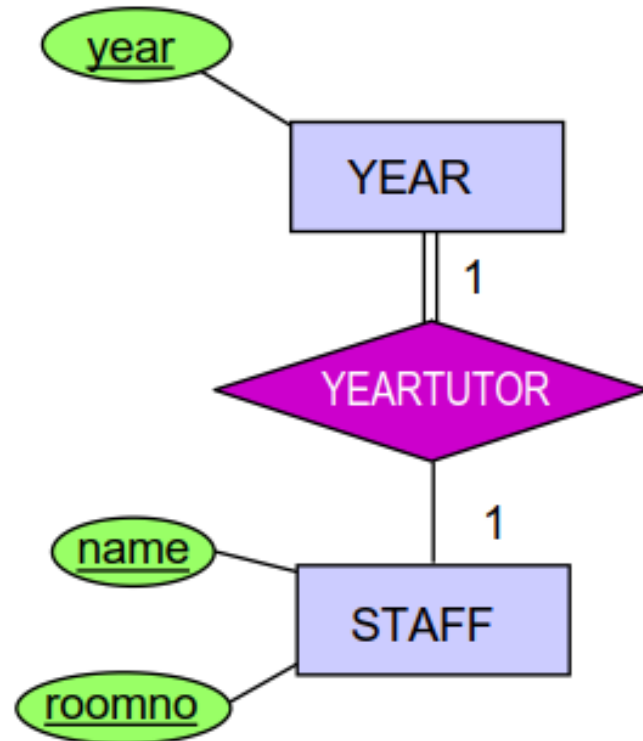
<u>studno</u>	givenname	familyname
s1	fred	jones
s2	mary	brown

Tabel StudentContact

<u>studno</u>	contact
s1	Mr. Jones
s1	Mrs Jones
s2	Bill Brown
s2	Mrs Jones
s2	Billy-Jo Woods

# DERAJAT RELASI 1-1

Relasi *one to one* menghubungkan 2 buah himpunan entitas, dipresentasikan dengan penambahan *field* relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas



Tabel Year

<u>year</u>	yeartutor

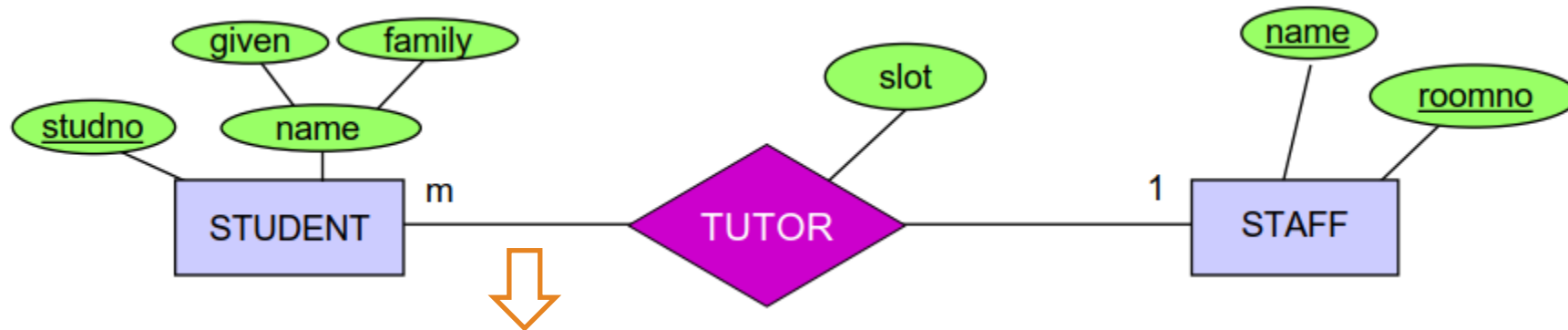
Tabel Staff

<u>name</u>	<u>roomno</u>	year



# DERAJAT RELASI 1-N

Relasi *one to many* yang menghubungkan 2 buah himpunan entitas, juga akan dipresentasikan dalam bentuk pemberian atribut key dari himpunan entitas pertama (berderajat 1) ke tabel yang mewakili himpunan entitas kedua (berderajat N). Atribut key dari himpunan entitas pertama ini menjadi atribut tambahan bagi himpunan entitas kedua



Tabel Staff

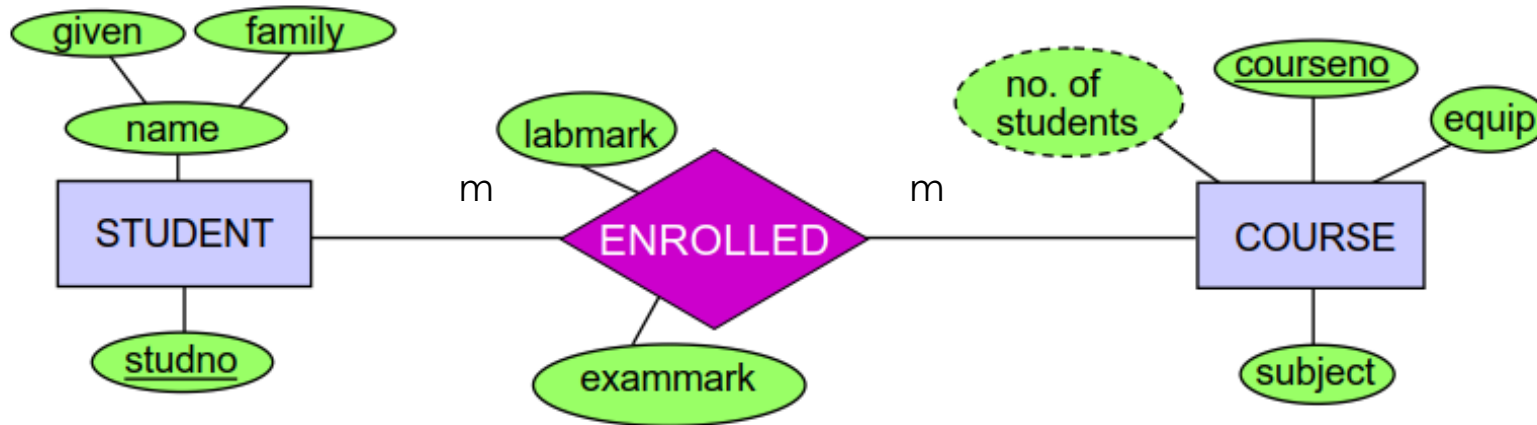
<u>name</u>	<u>roomno</u>

Tabel Student

<u>studno</u>	givenname	familyname	name	roomno	slot

# DERAJAT RELASI N-N

Relasi *many to many* yang menghubungkan 2 buah himpunan entitas, akan diwujudkan dalam bentuk tabel khusus yang memiliki *field (foreign key)* yang berasal dari *key-key* himpunan entitas yang dihubungkan



Tabel Student

<u>studno</u>	givenname	familyname

Tabel Course

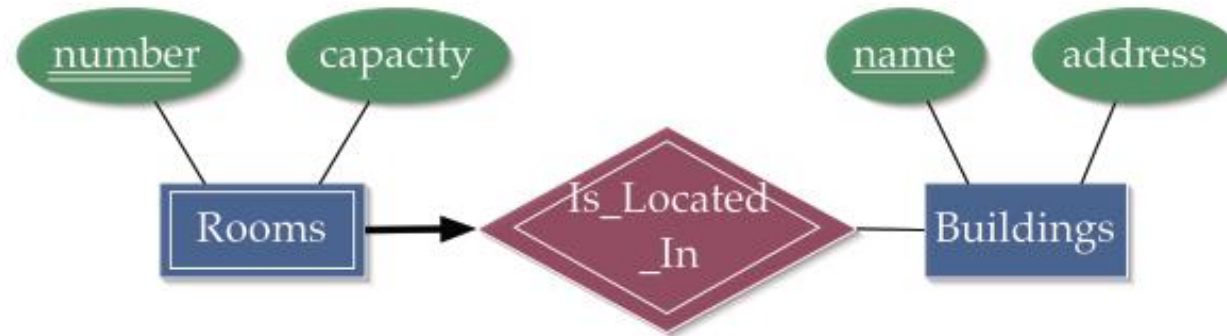
<u>courseno</u>	subject	equip

Tabel Student

<u>studno</u>	<u>courseno</u>	labmark	exammark

# ENTITAS LEMAH

Penggunaan Himpunan Entitas Lemah (*Weak Entity Sets*) dan Sub-Entitas dalam Diagram E-R diterapkan dalam bentuk tabel sebagaimana Himpunan Entitas Kuat (*Strong Entity Sets*)



Tabel Buildings

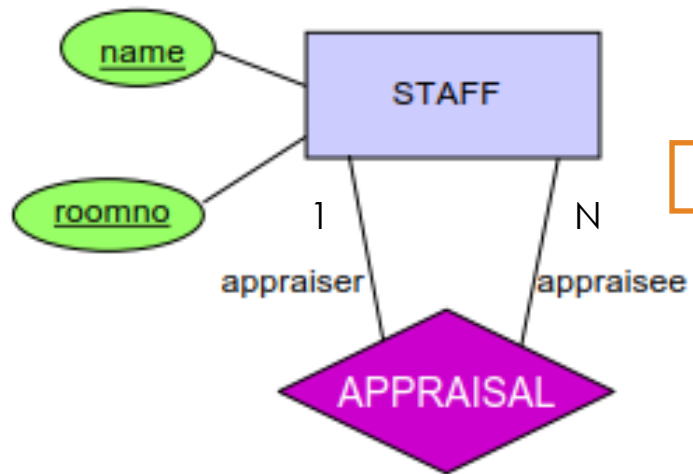
<u>name</u>	address

Tabel Rooms

<u>name</u>	number	capacity

# RELASI TUNGGAL (UNARY RELATION)

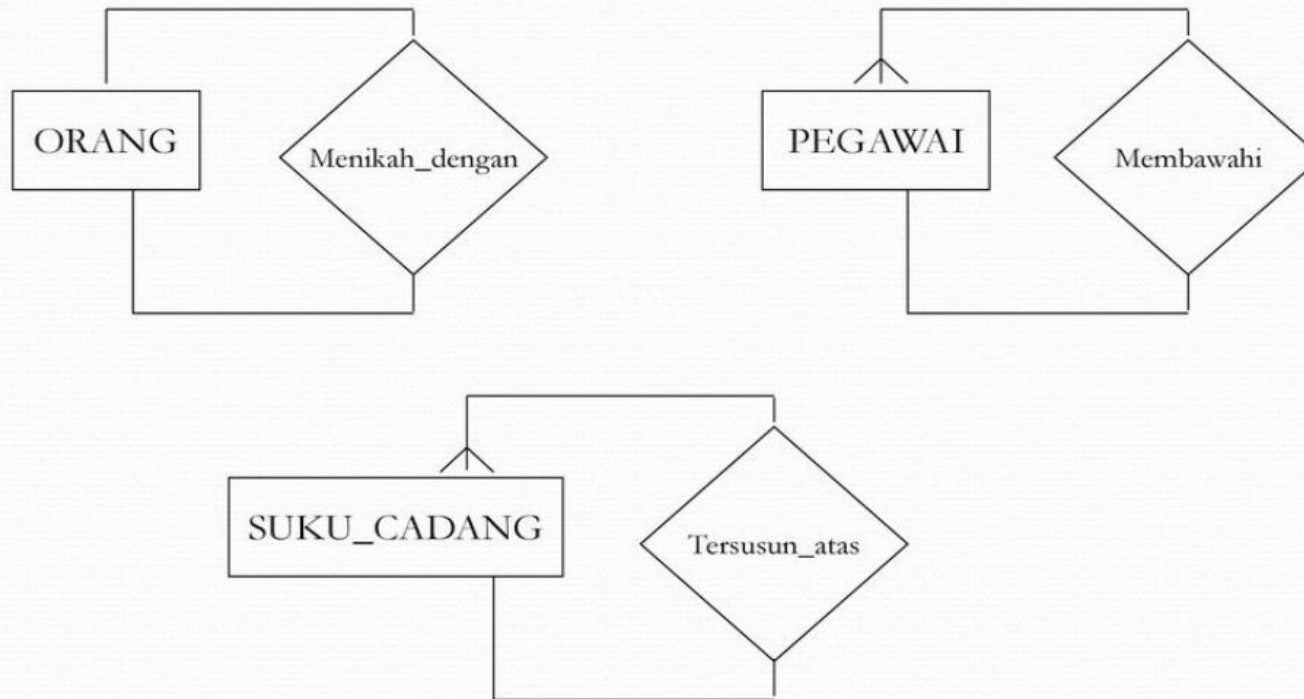
Penerapan Relasi Tunggal (Unary Relation) dari/ke himpunan entitas yang sama dalam Diagram E-R tergantung pada Derajat Relasinya. Unit Relasi Tunggal dengan Derajat Relasi satu-ke-banyak dapat diimplementasikan melalui penggunaan field key sebanyak dua kali lipat untuk fungsi yang berbeda.



Tabel Staff

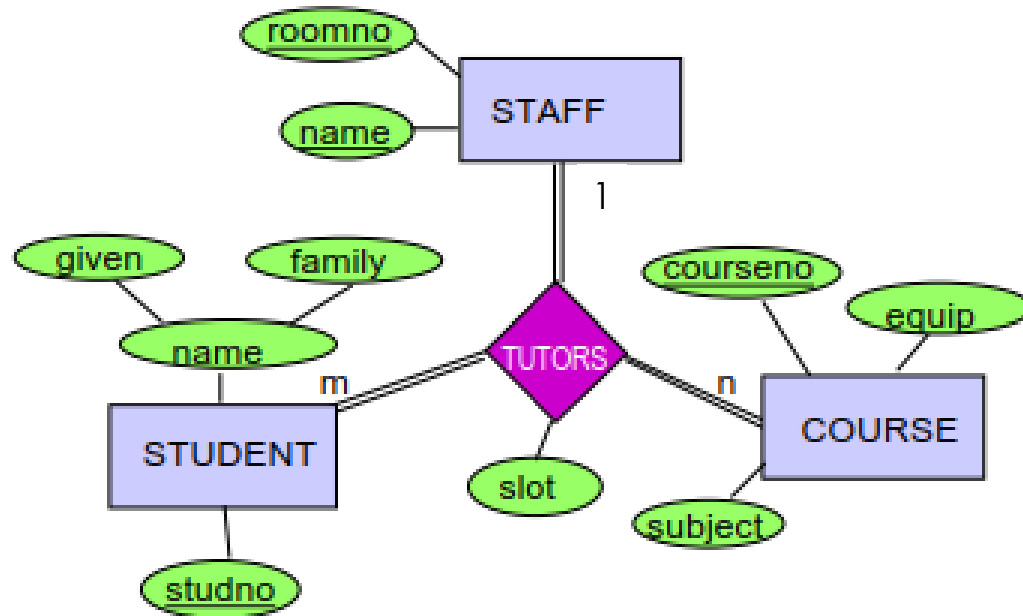
<u>name</u>	<u>roomno</u>	appraiserno

## Contoh Unary Relationship



# MULTI-ENTITAS(N-ARY RELATION)

Relasi multi entitas menghubungkan lebih dari dua himpunan entitas, untuk itu akan diterapkan sebuah tabel khusus. Namun jika dapat dipastikan bahwa hubungan antar entitas adalah 1-N maka cukup ditambahkan field pada entitas yang bersangkutan.



Tabel Students

<u>studno</u>	given name	family name

Tabel Course

<u>courseno</u>	subject	equip

Tabel Staff

<u>roomno</u>	<u>name</u>

Tabel Tutors

<u>roomno</u>	<u>nameStaff</u>	<u>studno</u>	<u>courseno</u>	slot

## RELASI GANDA (REDUDANT RELATION)

Relasi ganda diterapkan dengan cara yang sama dengan kardinalitasnya sesuai penjelasan sebelumnya

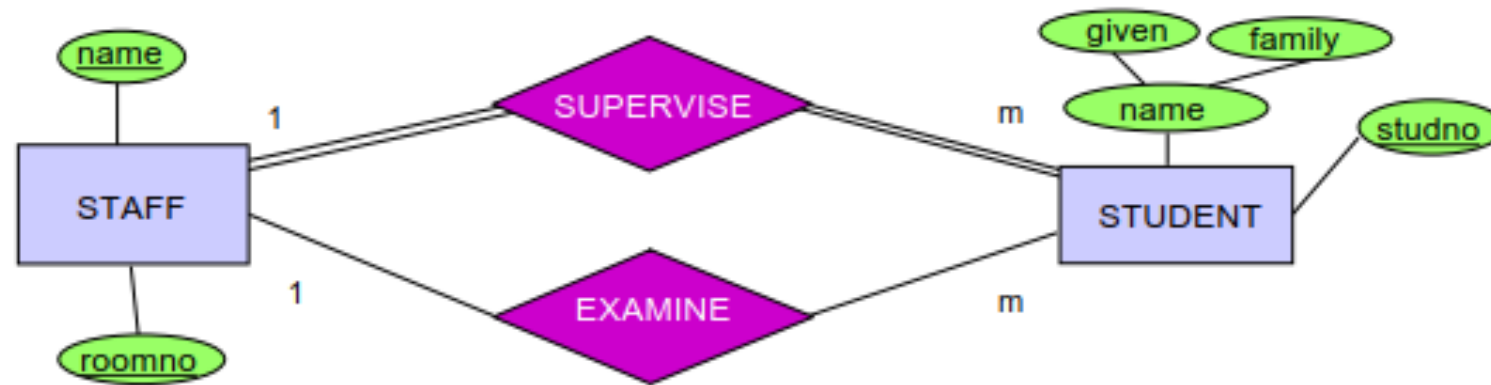


Table Staff

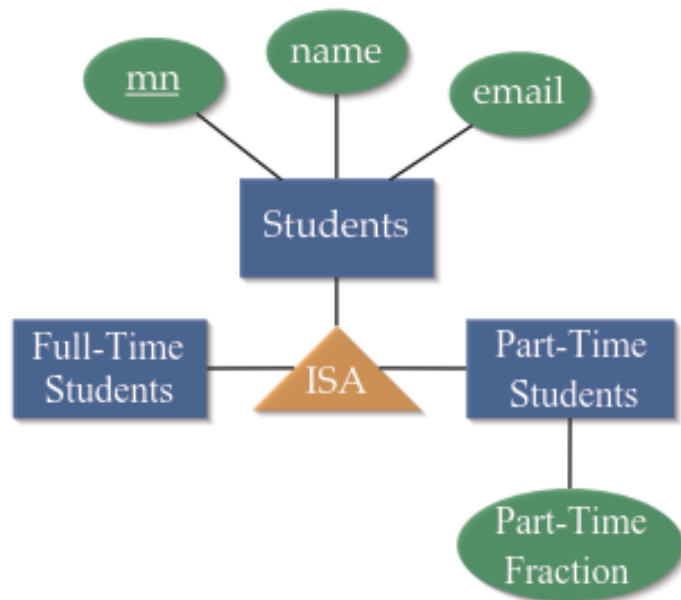
<u>name</u>	<u>roomno</u>

Tabel Student

<u>studno</u>	given	family	name	roomno

# SPEKIALISASI DAN GENERALISASI

Generalisasi menyusutkan jumlah entitas, sehingga hanya butuh satu entitas dengan penambahan field.  
Spesialisasi akan menghasilkan sejumlah himpunan entitas baru



Tabel Students

<u>mn</u>	name	email	part_frac



# MAPPING KE SKEMA RELASI (1)



Untuk melakukan **mapping (pemetaan)** dari skema ER Diagram ke skema relasi terdapat langkah-langkah yang harus diperhatikan.

## Langkah-langkah mapping:

1. Untuk **setiap entitas** 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.

# MAPPING KE SKEMA RELASI (2)



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**".

Apabila kedua partisipasi adalah **sama total**, maka kedua entitas tersebut boleh **digabung** menjadi satu skema relasi.

4. 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 (Many)**. 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.

# MAPPING KE SKEMA RELASI (3)

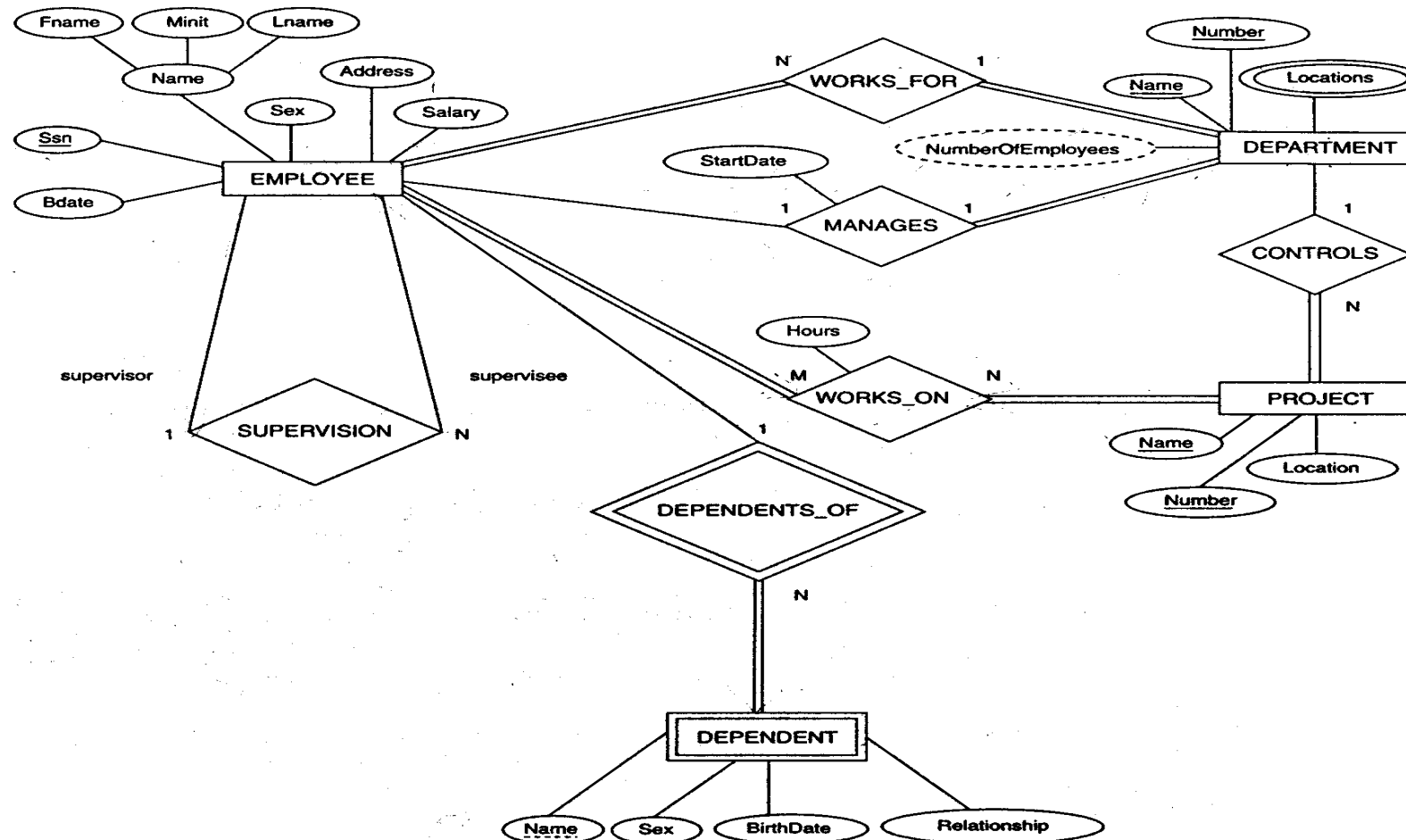


5. Untuk setiap relasi binary **M:N**, buatlah skema relasi baru **R** dengan atribut **seluruh simple attribute** yang terdapat pada relasi biner tersebut. Tambahkan **primary key** yang terdapat pada kedua sisi ke skema relasi R. Kedua **Foreign Key** yang didapat dari kedua sisi tersebut digabung menjadi satu membentuk **Primary Key** dari skema relasi R.
6. 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.
7. 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.

# MAPPING KE SKEMA RELASI (4)



Diagram Skema Konseptual / ER Diagram untuk Database COMPANY



# MAPPING KE SKEMA RELASI (5)



## EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

## DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

## DEPT\_LOCATIONS

<u>DNUMBER</u>	DLOCATION
----------------	-----------

## PROJECT

PNAME	<u>PNUMBER</u>	LOCATION	DNUM
-------	----------------	----------	------

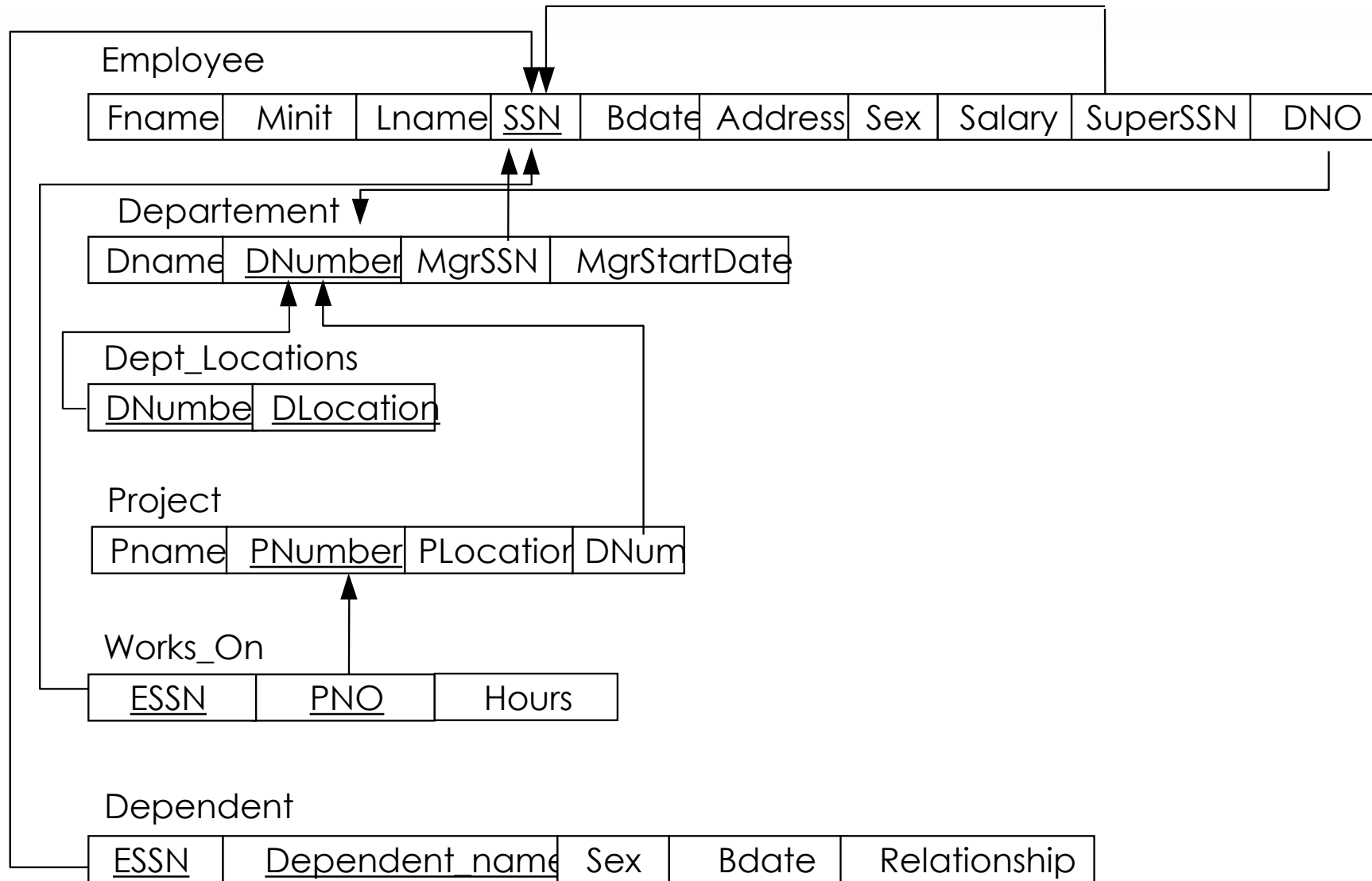
## WORKS ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

## DEPENDENT

<u>SSN</u>	<u>DEPENDENT_NAME</u>	SEX	BDATE	RELATIONSHIP
------------	-----------------------	-----	-------	--------------

# Mapping Skema ER Diagram dengan Referential Integrity Constraint



# Contoh Implementasi Data Table



## DEPARTMENT

dname	dnumber	mgrssn	mgrstartdate
Headquarters	1	333445555	22-May-78
administration	4	987654321	01-Jan-85
Research	5	888665555	19-Jun-71

## WORKS ON

## DEPARTMENT LOCATION

dnumber	dlocations
1	Houston
4	Stafford
5	Bellaire
5	Houston
5	Sugarland

essn	pno	hours
123456789	1	32.5
123456789	2	7.5
333445555	10	10
333445555	2	10
333445555	20	10
333445555	3	10
453453453	1	20
453453453	2	20
666884444	3	40
888665555	20	0
987654321	20	15
987654321	30	20
999887777	10	10
999887777	30	30

# Contoh Implementasi Data Table



## DEPENDENT

essn	dependent_name	sex	bdate	relationship
123456789	Alice	F	31-Dec-78	DAUGHTER
123456789	Elizabeth	F	05-May-57	SPOUSE
123456789	Michael	M	01-Jan-78	SON
333445555	Alice	F	05-Apr-76	DAUGHTER
333445555	Joy	F	03-May-48	SPOUSE
333445555	Theodore	M	25-Oct-73	SON
987654321	abner	M	29-Feb-32	SPOUSE

## EMPLOYEE

fname	minit	lname	ssn	bdate	address	sex	salary	superssn	dno
Ahmad	V	Jabbar	987987987	29-Mar-59	980 Dallas, Houston, TX	M	25000	987654321	4
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle, Spring, TX	F	25000	987654321	4
Franklin	T	Wong	333445555	08-Dec-45	638 Voss, Houston, TX	M	40000	888665555	5
James	E	Borg	888665555	10-Nov-27	450 Stone, Houston, TX	M	55000		1
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry, Bellaire, TX	F	43000	888665555	4
John	B	Smith	123456789	01-Sep-55	731 Fondren, Houston, TX	M	30000	333445555	5
Joyce	A	English	453453453	31-Jul-62	5631 Rice, Houston, TX	F			
Ramesh	K	Narayan	666884444	15-Sep-52	975 Fire Oak, Humble, TX	F			

## PROJECT

pname	pnumber	plocation	dnum
ProductX	1	Bellaire	5
Computerization	10	Stafford	4
ProductY	2	Sugarland	5
Reorganization	20	Houston	1
ProductZ	3	Houston	5
Newbenefits	30	Stafford	4

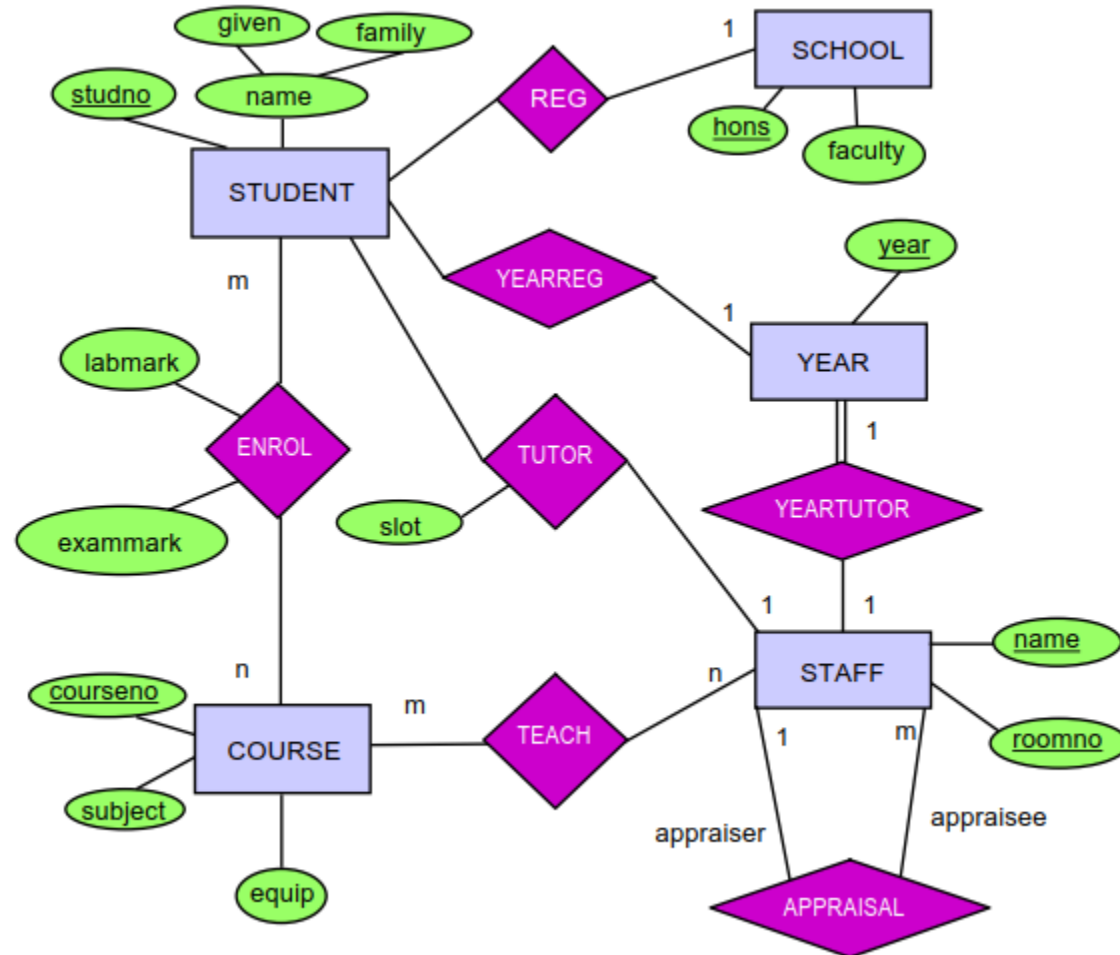




*Terima Kasih*

# LATIHAN 1

- Buatlah model relasional dari ERD berikut ini :

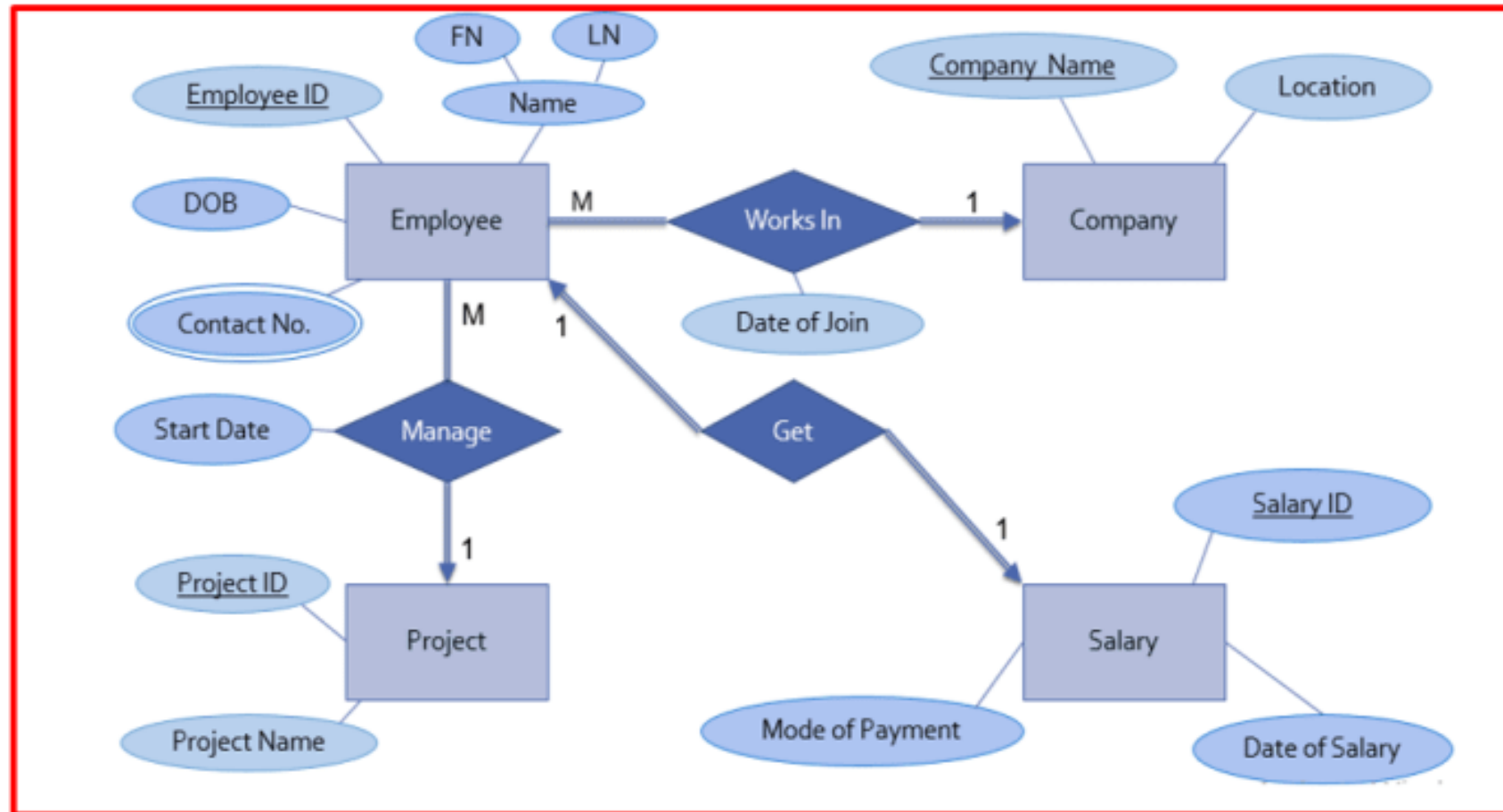




# JAWABAN LATIHAN 1

- STUDENT(studno, givenname, familyname, hons, tutor, tutorroom, slot, year)
- ENROL(studno, courseno, labmark, exammark)
- COURSE(courseno, subject, equip)
- STAFF(lecturer, roomno, appraiser, approom)
- TEACH(courseno, lecturer, lecroom )
- YEAR(year, yeartutor, yeartutorroom)
- SCHOOL(hons, faculty)

## LATIHAN 2



# JAWABAN LATIHAN 2



Employee( Employee ID , DOB , FN , LN , Salary ID )

EmployeeContact( Employee ID, Contact No)

Company( Company Name , Location, Employee ID, Date of Join )

Project( Project ID , Project Name , Employee ID, Start Date )

Salary( Salary ID , Mode of Payment, Date of Salary, Employee ID )



# REFERENSI

- Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, Program Studi Manajemen Informatika Politeknik Negeri Malang, 2012.
- indrajani. Pengantar Sistem Basis Data Case Study All in One. Elex Media Komputindo. 2014
- Fathansyah, “**Basisdata Revisi Kedua**”, Bandung: Informatika, 2015.
- <http://www.dcs.warwick.ac.uk/~hugh/TTM/The-Relational-Model.pdf>
- <http://infolab.stanford.edu/~ullman/focs/ch08.pdf>
- <http://www.inf.unibz.it/~nutt/IDBs1011/IDBSlides/5-db-erToRel.pdf>
- <http://teknikinformatika.fasilkom.mercubuana.ac.id/wp-content/uploads/2015/03/3.-Modul-3-Model-Data-Relasional.pdf>