

# PERANCANGAN SISTEM DATABASE

## 1. PERANCANGAN DATABASE

Perancangan Database adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem.

Perancangan sistem terjadi pada dua tingkat, yaitu :

**Pada tingkat pertama**, perencanaan sistem, analisis dan rancangan umum dilaksanakan untuk menetapkan kebutuhan pemakai. Tingkat perancangan database ini melibatkan tahap *front-end*, bebas dari perancangan database tertentu atau Database Management System (DBMS).

**Pada tingkat kedua**, rancangan umum, seperti diagram entitas relasi tingkat tinggi, ditransformasikan (atau didekomposisikan) ke dalam perancangan database rinci untuk sebuah DBMS tertentu yang akan digunakan untuk mengimplementasikan sistem total.

Tiga model database yang cukup dikenal adalah :

- Model Hierarkikal
- Model Jaringan
- Model Relasional

Pada masa lalu banyak penjual (*vendors*) menawarkan Database Management Systems (DBMS) yang berdasarkan pada Model Hierarkikal dan Model Jaringan. Saat ini Model Relasional adalah dominan. Karena itu hampir semua penjual perangkat lunak database menawarkan produk perangkat lunak Relational Database Management Systems (RDBMS).

RDBMS dibuat dengan struktur tiga skema, Struktur lapisan ini mendefinisikan data perusahaan pada tingkat yang berbeda.

**Skema Eksternal** mendefinisikan bagaimana pemakai mengakses dan melihat output dari RDBMS, bebas dari bagaimana data disimpan atau diakses secara fisik. Akses dan manipulasi seperti ini dilaksanakan oleh pemakai dengan memperkerjakan bahasa prosedural, seperti COBOL atau bahasa query, seperti Structured Query Language (SQL), bahasa standar yang diakui untuk RDBMS.

**Skema Konseptual** yang mendefinisikan model database relasional terdiri dari sekumpulan tabel yang dinormalisasi. Skema konseptual adalah rancangan dari database yang merupakan subyek utama dari bab ini.

**Skema Internal** terdiri dari organisasi fisik dari data (mis. sekuensial, indeks sekuensial, langsung) dalam hal struktur fisik data dan metode-metode pengaksesan dari sistem operasi komputer.

### APAKAH DATABASE RELASIONAL ITU ?

Model relasional berdasarkan teori himpunan matematik. Struktur didefinisikan dengan Tabel. Dalam istilah matematika, tabel disebut sebagai Relasi. Profesional sistem sering menggunakan istilah “tabel” dan “relasi” secara bergantian.

Tiap tabel dalam model relasional dikomposisikan dari baris dan kolom. Kolom disebut Atribut. Nilai untuk sebuah atribut harus dipilih dari sekelompok nilai yang dinamakan Domain. Karena banyak kolom dalam tabel yang sama dapat diidentifikasi atas domain yang sama, maka nama atribut didefinisikan untuk tiap kolom. Tiap nama atribut dalam sebuah relasi harus unik. Urutan kiri ke kanan dari kolom tidak penting. Urutan dari baris juga tidak penting. Perpotongan dari suatu baris dan kolom berisi sebuah nilai tunggal.

### Sifat-sifat Tabel :

1. Duplikasi baris tidak diperbolehkan. Untuk melaksanakan sifat ini, harus terdapat paling sedikit satu atribut atau kombinasi beberapa atribut yang mengidentifikasi secara unik tiap baris dari tabel. Atribut atau kombinasi beberapa atribut yang melaksanakan tugas ini disebut **Kunci Primer** (*Primary Key*). Contoh : Nomor\_Mahasiswa, adalah kunci primer yang mengidentifikasi tiap mahasiswa secara unik.
2. Database relasional adalah nilai *Primary Key* tidak boleh mempunyai duplikat atau NIL (NULL, yaitu nilai tidak diketahui).
3. Keterhubungan (*relationship*) antara dua tabel. Jika Tabel R2 mempunyai sebuah **Kunci Asing** (*Foreign Key*) yang cocok dengan kunci primer dari Tabel R1, maka untuk setiap nilai *Foreign Key* harus terdapat sebuah nilai kecocokan dari *Primary Key*, atau nilai *Foreign Key* harus nil.

## Menggunakan Structures Query Language (SQL)

SQL adalah bahasa standar database yang digunakan untuk query, manipulasi dan memperbaiki RDBMS. Karena semakin banyak organisasi yang memutuskan untuk mengkonsolidasikan database mereka ke dalam sistem seluas usaha, pengetahuan mengenai SQL akan menjadi kebutuhan untuk para perancang database.

## 2. ENTITY RELATIONSHIP DIAGRAM (ERD)

### Model Entity Relationship

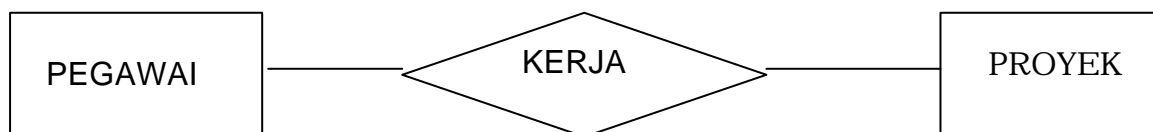
Adalah suatu penyajian data dengan menggunakan Entity dan Relationship.

### Entity

- Entity adalah obyek yang dapat dibedakan dalam dunia nyata
- Entity set adalah kumpulan dari entity yang sejenis
- Entity set dapat berupa :
  - Obyek secara fisik : Rumah, Kendaraan, Peralatan
  - Obyek secara konsep : Pekerjaan , Perusahaan, Rencana

### Relationship

- Relationship adalah hubungan yang terjadi antara satu atau lebih entity.
- Relationship set adalah kumpulan relationship yang sejenis.

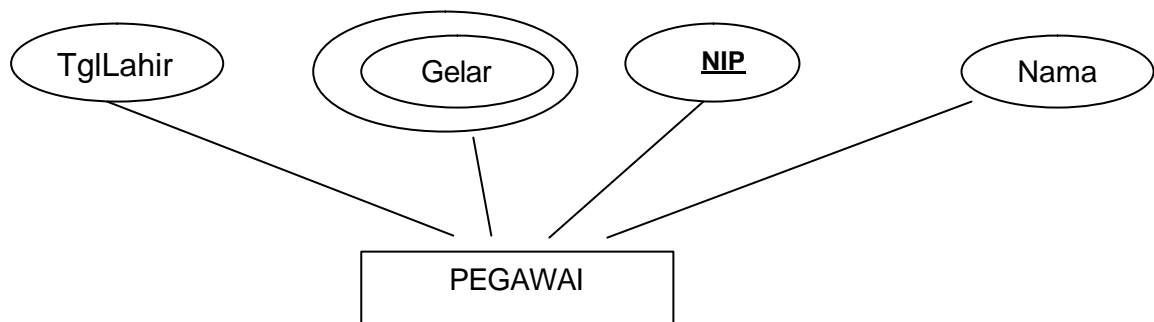


### Atribut

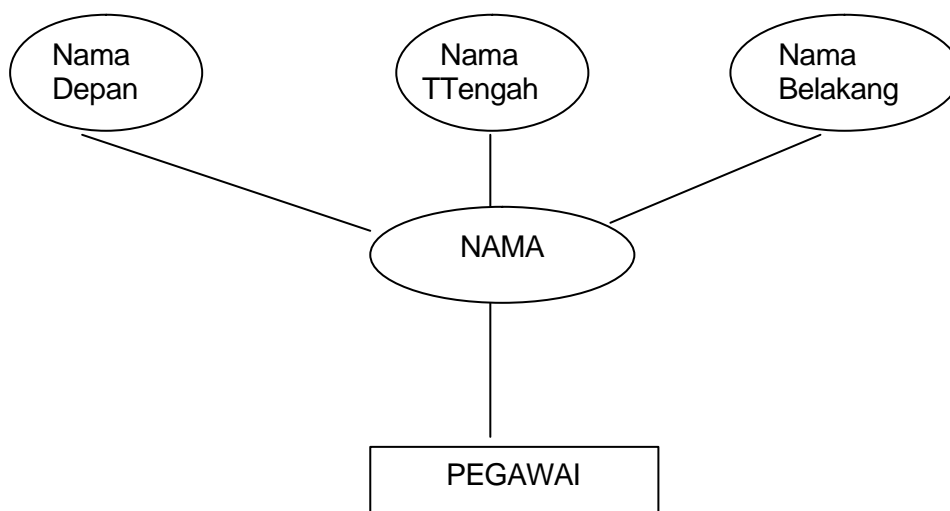
- Atribut adalah karakteristik dari entity atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tersebut.
- Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

### Jenis-jenis atribut :

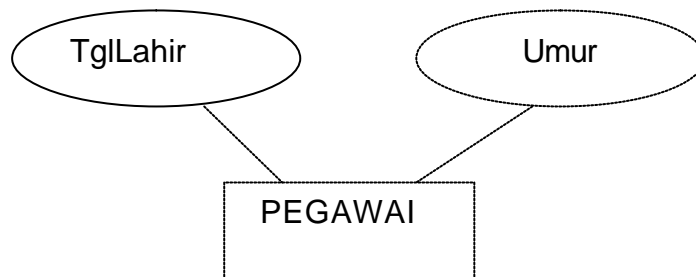
- **Key**  
Atribut yang digunakan untuk menentukan suatu entity secara unik.
- **Atribut Simple**  
Atribut yang bernilai tunggal.
- **Atribut Multivalued**  
Atribut yang memiliki sekelompok nilai untuk setiap instan entity.



- **Atribut Composite**  
Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.



- **Atribut Derivatif**  
Suatu atribut yang dihasilkan dari atribut yang lain.



### Derajat dari relationship

Menjelaskan jumlah entity yang berpartisipasi dalam suatu relationship.

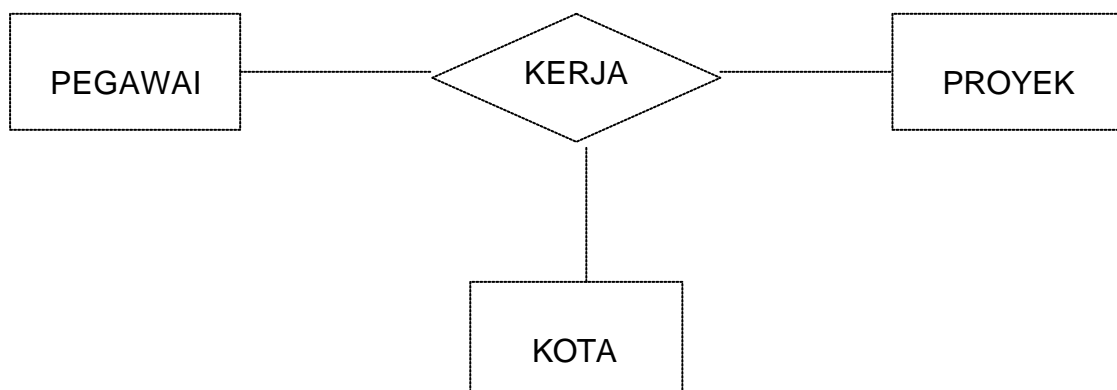
#### Unary Degree (Derajat Satu)



#### Binary Degree (Derajat Dua)



#### Ternary Degree (Derajat Tiga)

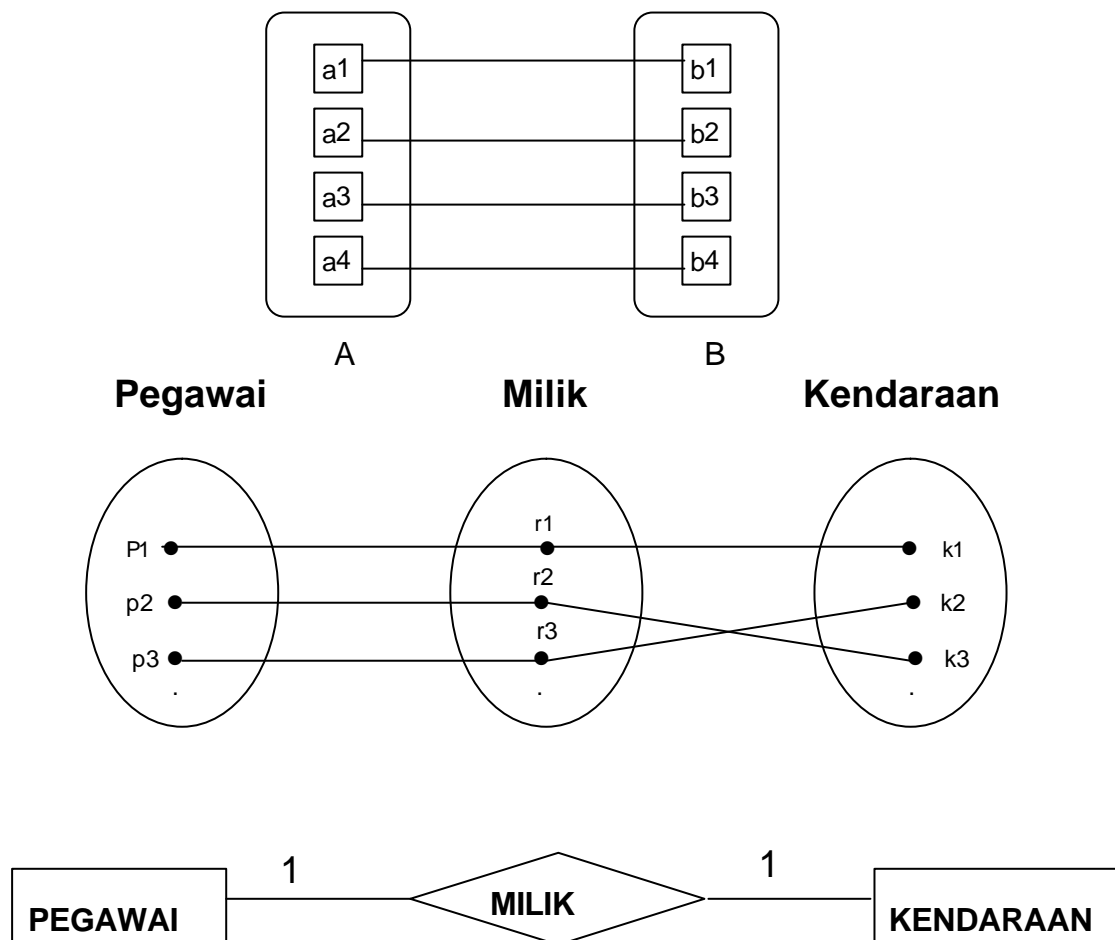


### Cardinality Ratio Constraint

- Menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.
- Jenis Cardinality Ratio

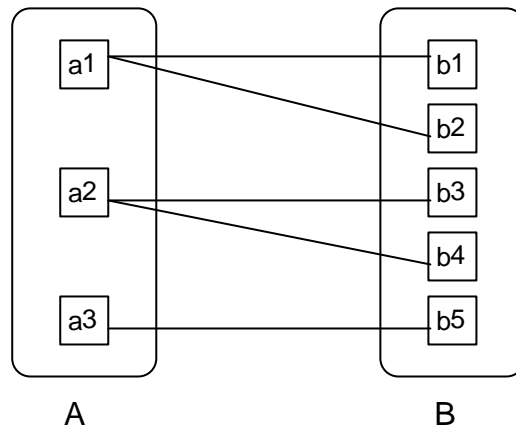
#### 1 : 1 (One-To-One)

Sebuah entity A diasosiasikan pada sebuah entity B, dan sebuah entity B diasosiasikan dengan paling banyak sebuah entity A.



#### 1 : N (One-To-Many)

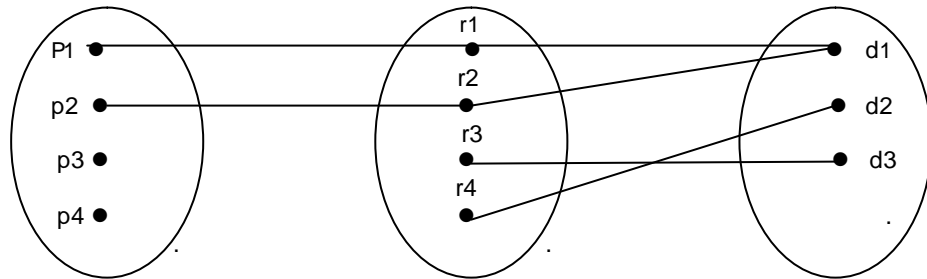
Sebuah entity A diasosiasikan dengan sejumlah entity B, tetapi entity B dapat diasosiasikan paling banyak satu entity A.



**Pegawai**

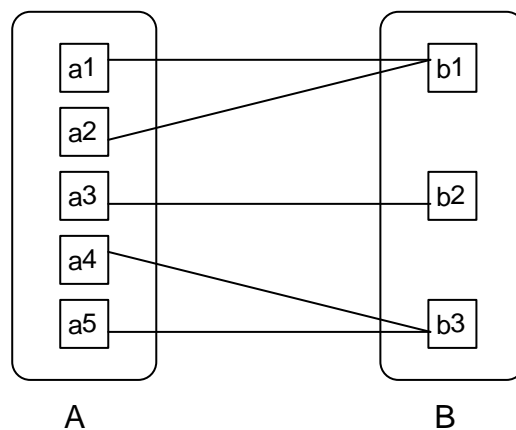
**Kerja**

**Departemen**



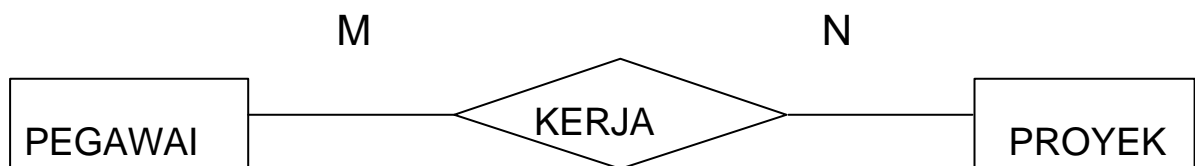
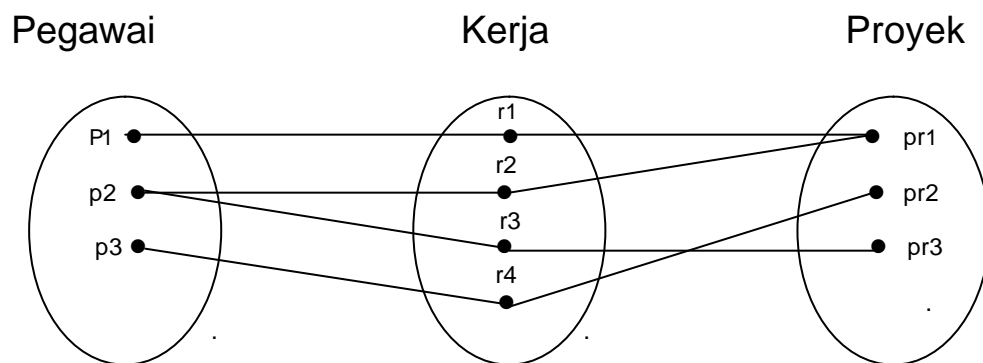
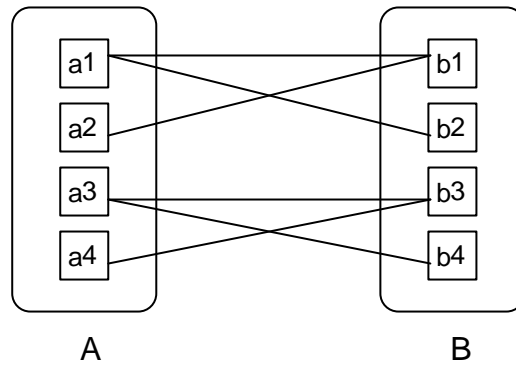
**N : 1 (Many-To-One)**

Suatu entity A dapat diasosiasikan dengan paling banyak sebuah entity B, tetapi entity B dapat diasosiasikan dengan sejumlah entity di A.



### M : N (Many-To-Many)

Suatu entity A dapat diasosiasikan dengan sejumlah entity B dan entity B dapat diasosiasikan dengan sejumlah entity di A.



### Participation Constraint

Menjelaskan apakah keberadaan suatu entity tergantung pada hubungannya dengan entity lain .

Terdapat 2 macam Participation Constraint :

#### Total Participation

Keberadaan suatu entity tergantung pada hubungannya dengan entity lain.





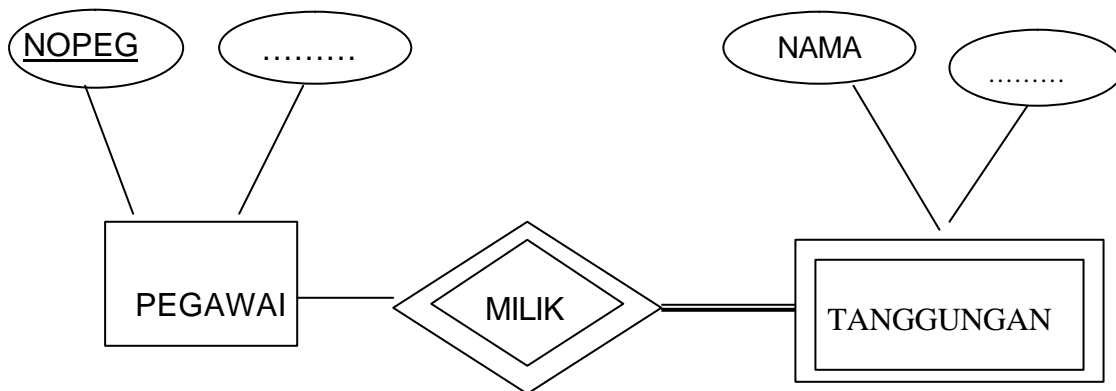
**Partial Participation**

Keberadaan suatu entity tidak tergantung pada hubungannya dengan entity lain.



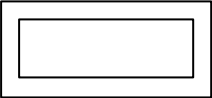
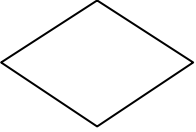
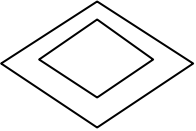

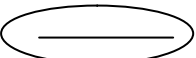

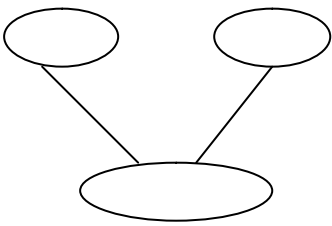

**Weak Entity**

- Weak Entity adalah suatu Entity dimana keberadaan dari entity tersebut tergantung dari keberadaan entity lain.
- Entity yang merupakan induknya disebut *Identifying Owner* dan relationshipnya disebut *Identifying Relationship*.
- Weak Entity selalu mempunyai Total Participation constraint dengan Identifying Owner.



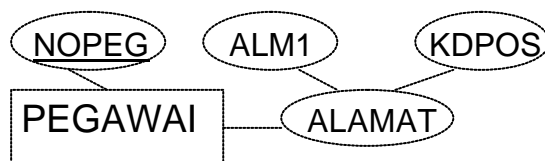
**Simbol-simbol ER-Diagram**

<u>Notasi</u>	<u>Arti</u>
1. 	1. Entity

- 2.  2. Weak Entity
- 3.  3. Relationship
- 4.  4. Identifying Relationship
- 5.  5. Atribut
- 6.  6. Atribut Primary Key
- 7.  7. Atribut Multivalued
- 8.  8. Atribut Composite
- 9.  9. Atribut Derivatif

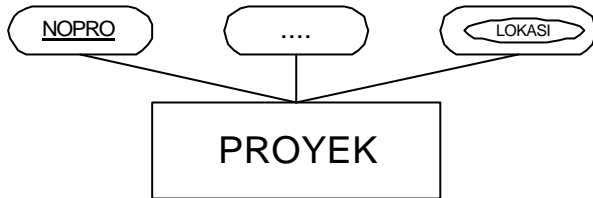
**Transformasi dari ERD ke Database Relasional**

1. Setiap tipe Entity dibuat suatu relasi yang memuat semua atribut simple, sedangkan untuk atribut composite hanya dimuat komponen-komponennya saja.



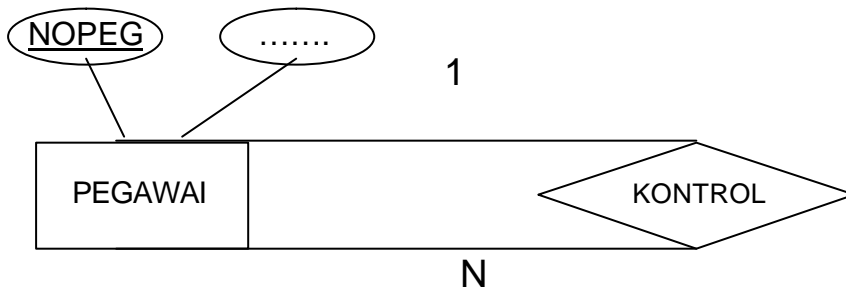
PEGAWAI (NOPEG, ALM1, KDPOS, .....)

2. Setiap relasi yang mempunyai atribut multivalued, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key dari relasi tersebut dengan atribut multivalued.



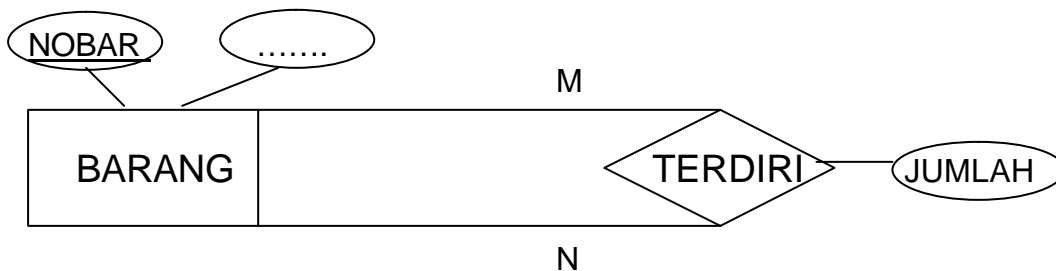
LOKPR(NOPRO, LOKASI)

3. Setiap Unary Relationship 1:N, pada relasi perlu ditambahkan suatu foreign key yang menunjuk ke nilai primary keynya.



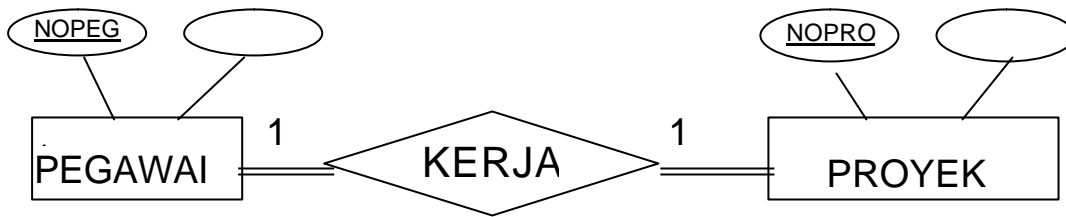
PEGAWAI (NOPEG, ....., SUPERVISOR-ID)

4. Setiap Unary Relationship M:N, buatlah relasi baru dimana primary keynya merupakan gabungan dari dua atribut dimana keduanya menunjuk ke primary key relasi awal dengan penamaan yang berbeda.



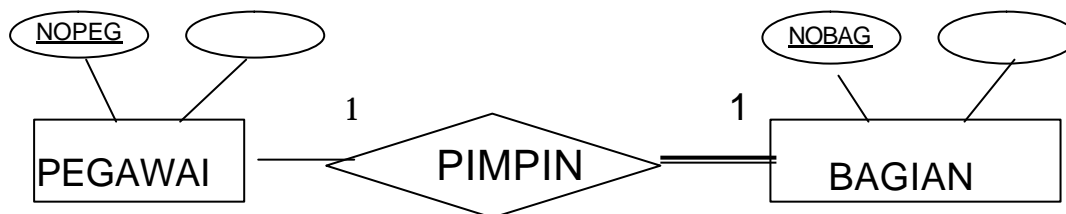
KOMBAR (NOBAR, NOKOMP , JUMLAH)

5. Setiap Binary Relationship 1:1, dimana Participation Constraint keduanya total, buatlah suatu relasi gabungan dimana Primary Keynya dapat dipilih salah satu.



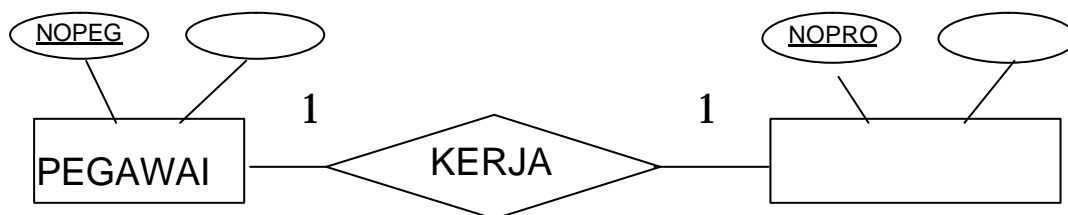
PEGAWAI (NOPEG, ... , NOPRO, ...).

6. Setiap Binary Relationship 1:1 dan salah satu Participation Constraintnya Total, maka Primary Key pada relasi yang Participation Constraintnya Partial menjadi Foreign Key pada relasi yang lainnya.



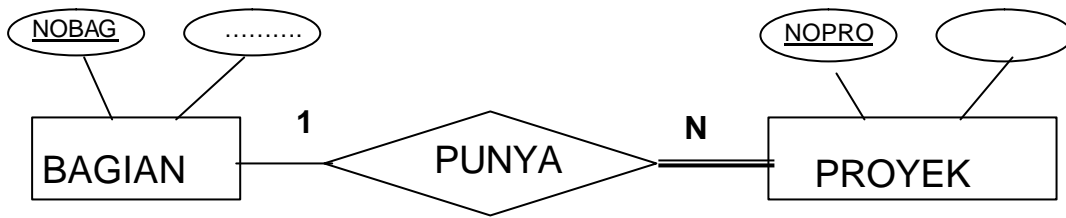
BAGIAN (NOBAG, ... , MANAGER)

7. Setiap Binary Relationship 1:1, dimana kedua Participation Constraintnya partial, maka selain kedua relasi perlu dibuat relasi baru yang berisi Primary Key gabungan dari Primary Key kedua tipe Entity yang berelasi.



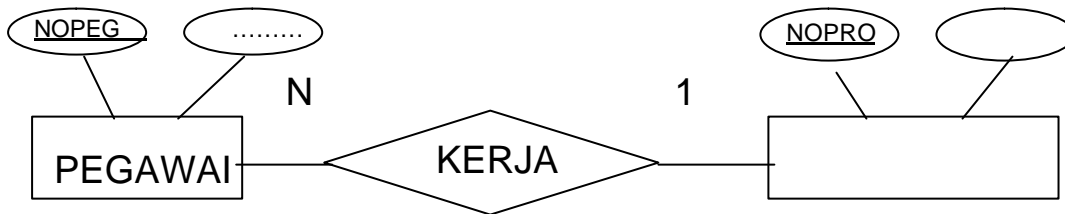
PEKERJAAN ( NOPEG, NOPRO, ...)

8. Setiap Binary Relationship 1 : N, dimana tipe Entity yang bersisi N mempunyai Participation Constraint Total, maka Primary Key pada relasi yang bersisi 1 dijadikan Foreign Key pada relasi yang bersisi N.



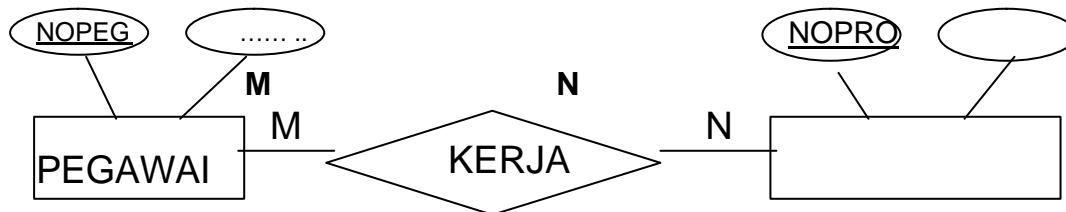
PROYEK (NOPRO, ... , NOBAG)

9. Setiap Binary Relationship 1 : N, dimana tipe Entity yang bersisi N mempunyai Participation Constraint partial, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key kedua tipe Entity yang berelasi.



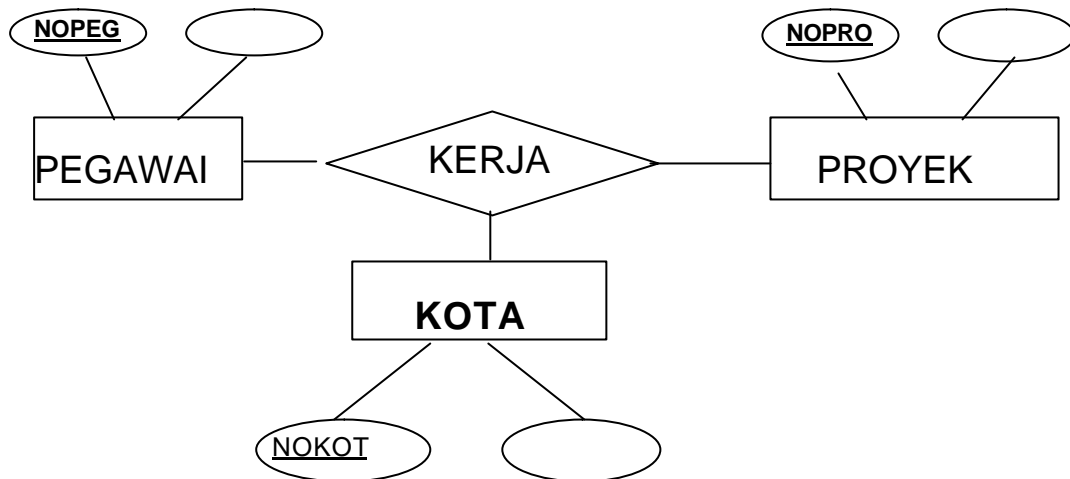
PEKERJAAN (NOPEG, NOPRO, .....

10. Setiap Binary Relationship M:N, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key kedua tipe Entity yang berelasi.



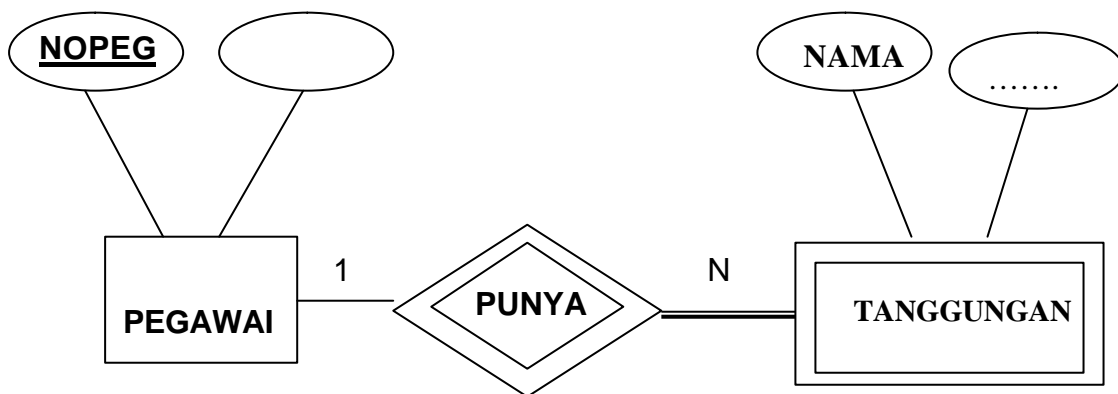
PEKERJAAN (NOPEG, NOPRO, .....

11. Setiap Ternary Relationship, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key ketiga tipe Entity yang berelasi.



PEKERJAAN ( NOPEG, NOPRO , NOKOT)

12. Setiap tipe Weak Entity, dibuat suatu relasi yang memuat semua atributnya dimana Primary Keynya adalah gabungan dari Partial Key dan Primary Key dari relasi induknya (identifying owner).



TANGGUNGAN (NOPEG, NAMA, ....)

**Hasil Transformasi dari Diagram ER ke Database Relasional :**

Skema Database

PEGAWAI	( <u>NOPEG</u> , NAPEG, ALM1, KDPOS, TGLLAH, UMUR, SUPERVISOR-ID, NOBAG)
BAGIAN	( <u>NOBAG</u> , NABAG, LOKASI, MANAGER)
PROYEK	( <u>NOPRO</u> , NAPRO, NOBAG)
LOKPR	( <u>NOPRO</u> , LOKAPR)
PEKERJAAN	( <u>NOPEG</u> , <u>NOPRO</u> , JAM)
TANGGUNGAN	( <u>NOPEG</u> , <u>NAMA</u> , JNKELT, HUBUNGAN)